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

Hands on Embedded ML
(Vision and Audio)

Brian Plancher
Harvard John A. Paulson School of Engineering and Applied Sciences
brianplancher.com

Quick Disclaimer:

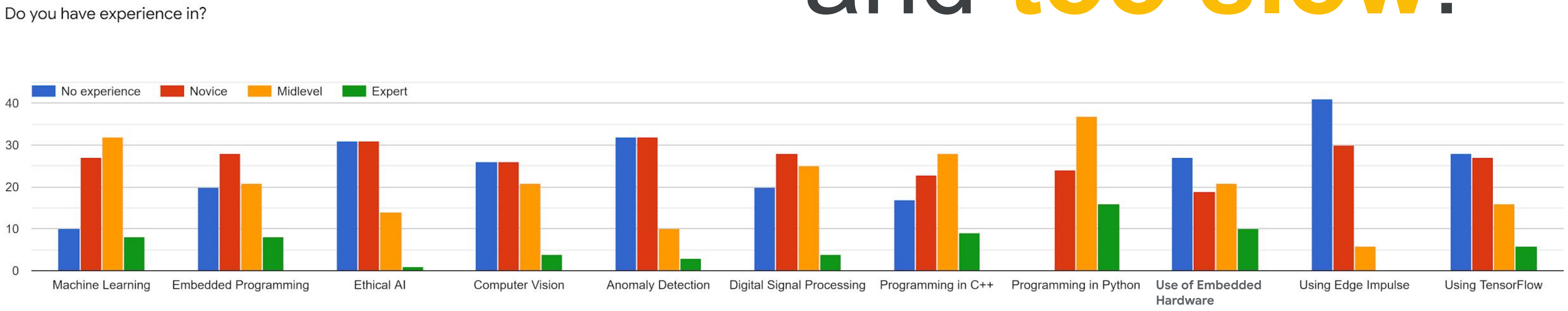
Today will be **both too fast**

and **too slow**!

Quick Disclaimer:

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

and **too slow**!



By the end of today: **Hands-on Keyword Spotting**

We will explore the **science** behind KWS and **collect data** and **train** our own custom model to recognize “yes” vs. “no” using **Edge Impulse**

Today's Agenda

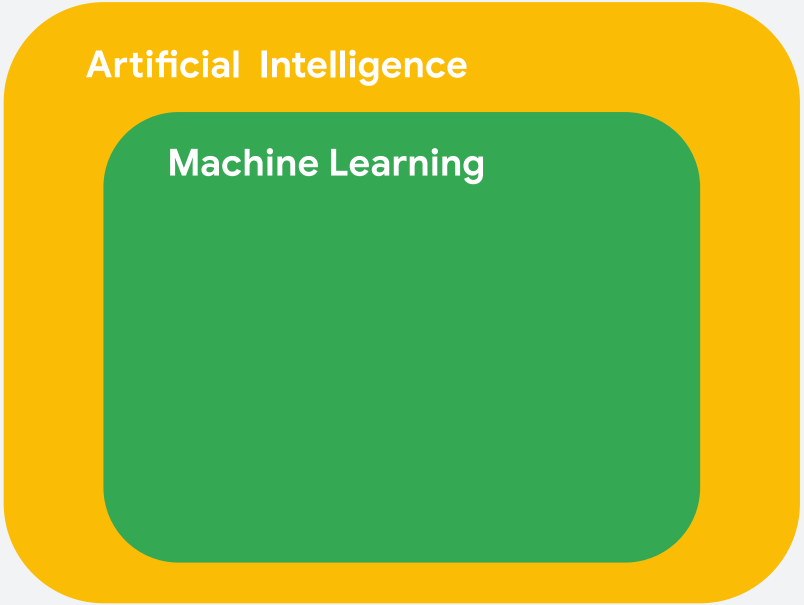
- Deep ML Background
- Hands-on Computer Vision: Thing Translator
- The Tiny Machine Learning Workflow
- Keyword Spotting (KWS) Data Collection
- KWS Preprocessing and Training
- Deployment Challenges and Opportunities for Embedded ML
- Summary

Today's Agenda

- Deep ML Background
  - **How does (Deep) Machine Learning Work?**
  - Exploring Deep ML through Computer Vision
- Hands-on Computer Vision: Thing Translator
- The Tiny Machine Learning Workflow
- Keyword Spotting (KWS) Data Collection
- KWS Preprocessing and Training
- Deployment Challenges and Opportunities for Embedded ML
- Summary

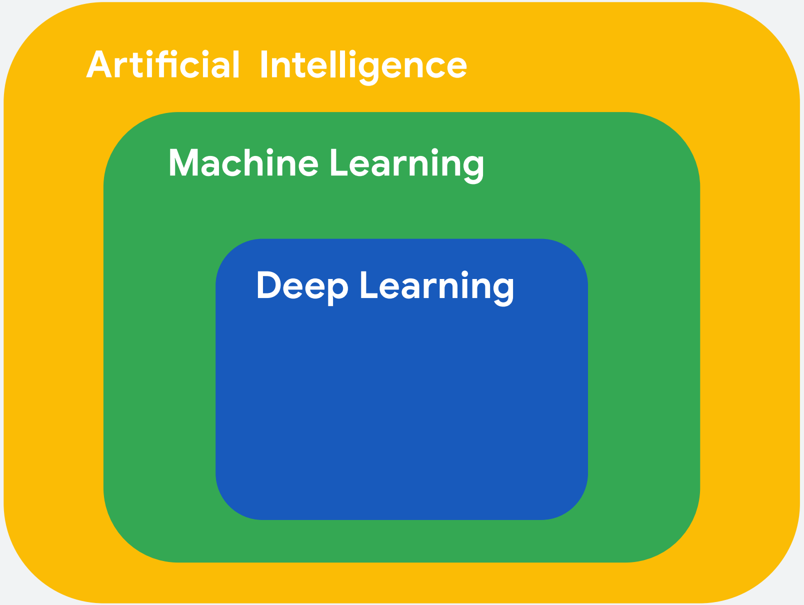
What is Machine Learning?

1. **Machine Learning** is a subfield of **Artificial Intelligence** focused on developing algorithms that learn to **solve problems by analyzing data for patterns**

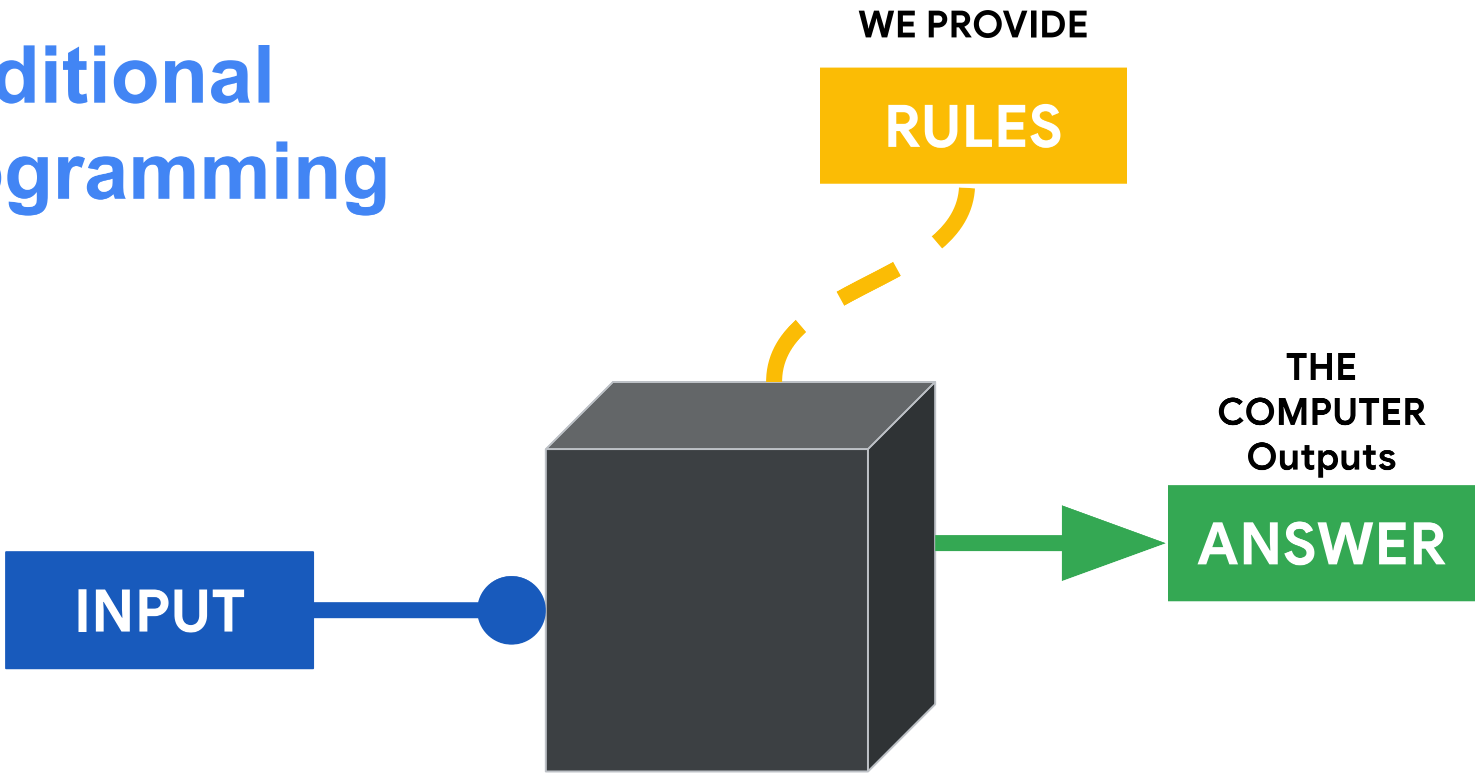


What is (Deep) Machine Learning?

1. Machine Learning is a subfield of Artificial Intelligence focused on developing algorithms that learn to solve problems by analyzing data for patterns
2. **Deep Learning** is a type of **Machine Learning** that leverages **Neural Networks** and **Big Data**



Traditional Programming



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

```
if (speed < 4):
     then walking
```

```
if (speed < 4):
     then walking

else:
     running
```

data we can gather

input: **speed**

Write a **rule**

extend the rule

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

```
if (speed < 4):
      then walking
```

```
if (speed < 4):
      then walking


else:
      running
```

```
if (speed < 4):
      then walking


else if (speed < 12):
      then running
else:
      biking
```

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

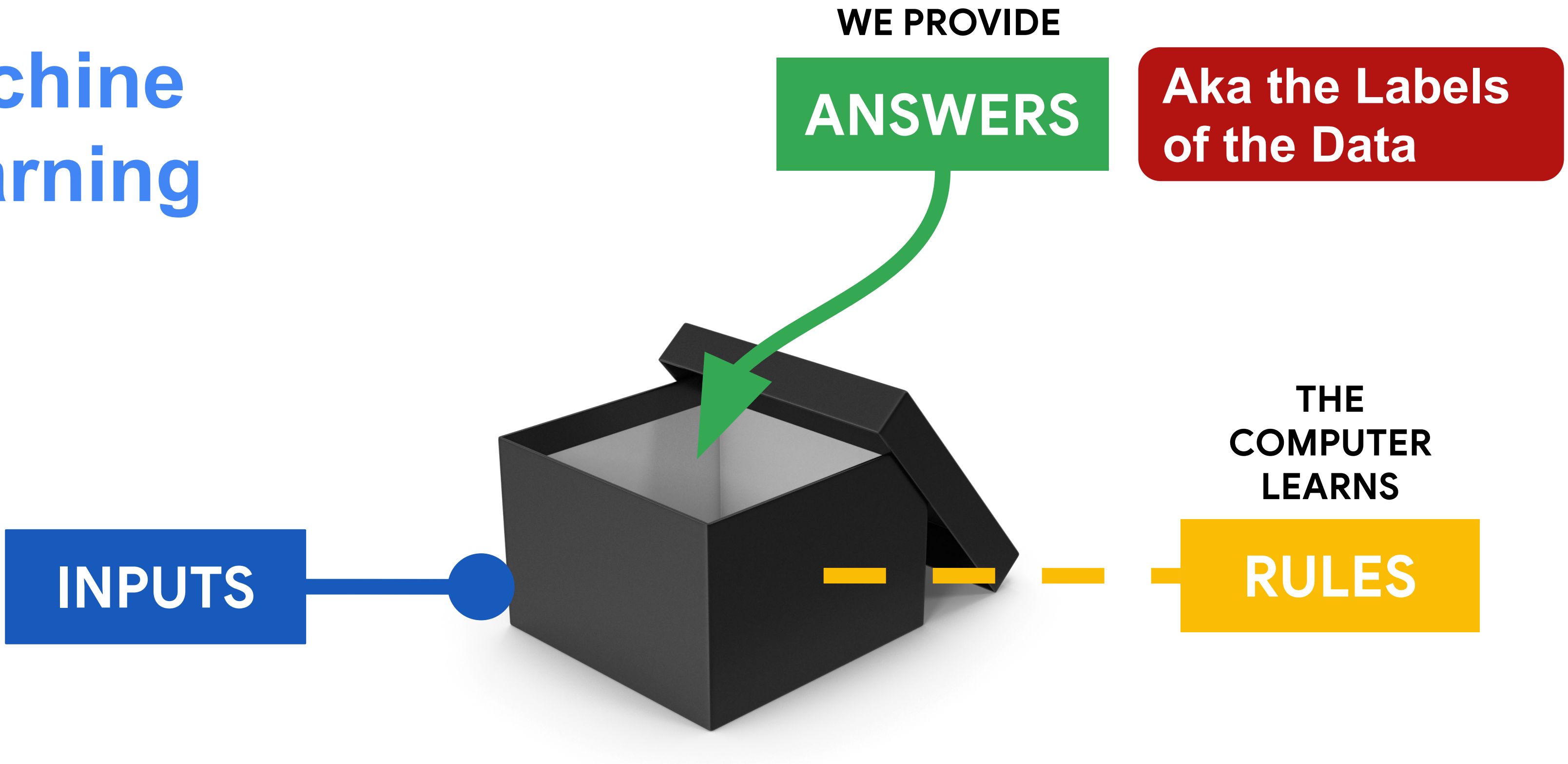
```
if (speed < 4):
    then walking
```

```
if (speed < 4):
    then walking

else:
    running
```

```
if (speed < 4):
    then walking

else if (speed < 12):
    then running
else:
    biking
```

?? **WHAT IS THIS** ??

Machine Learning



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

01010101001000110101
01010100101001001010
10101011010100101001

11110101001001010101
01010010100101010100
11010110010101001111

00001110101110101101
01010111101011010101
11010111111001001011

01111110101110101010
10101110101011010101
11111111100100001110

walking | **running** | **biking** | **golfing**

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

01010101001000110101
01010100101001001010
1010101101010010**1001**

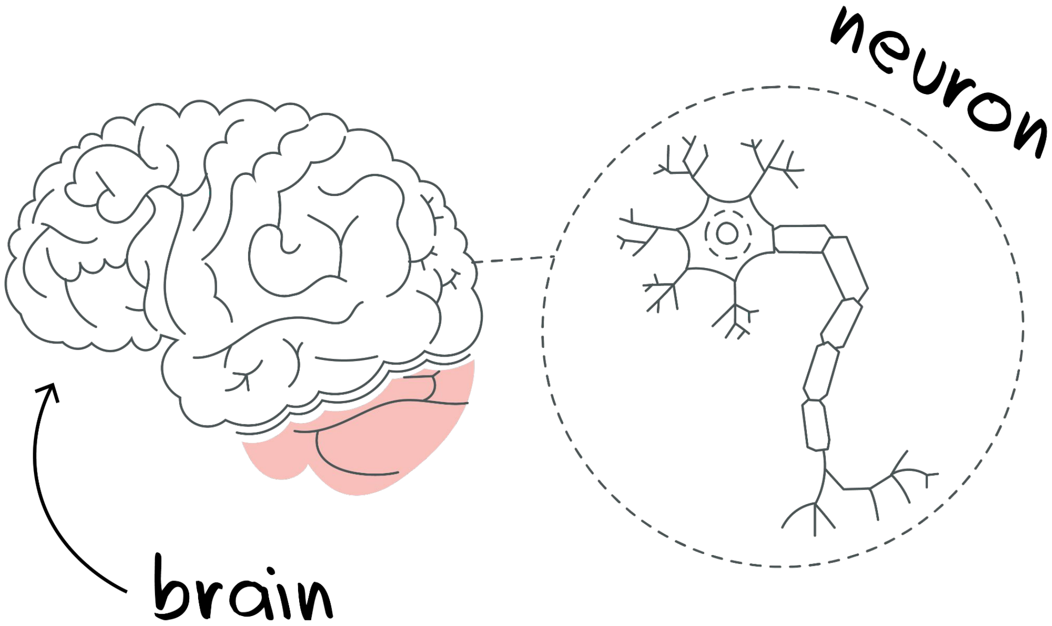
11110101001001010101
01010010100101010100
1101011001010100**1111**

00001110101110101101
01010111101011010101
1101011111100100**1011**

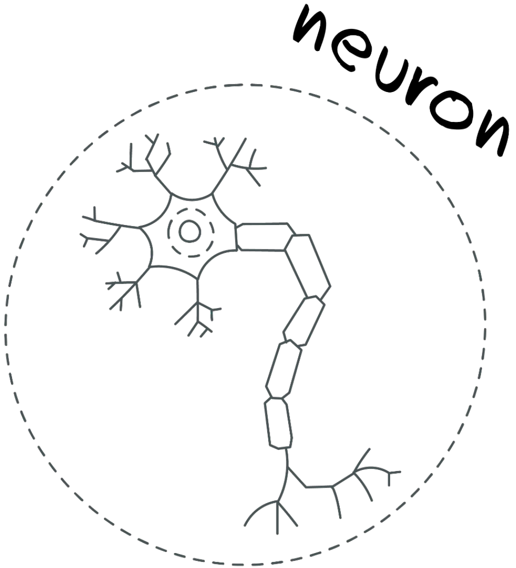
01111110101110101010
10101110101011010101
1111111110010000**1110**

walking **running** **biking** **golfing**

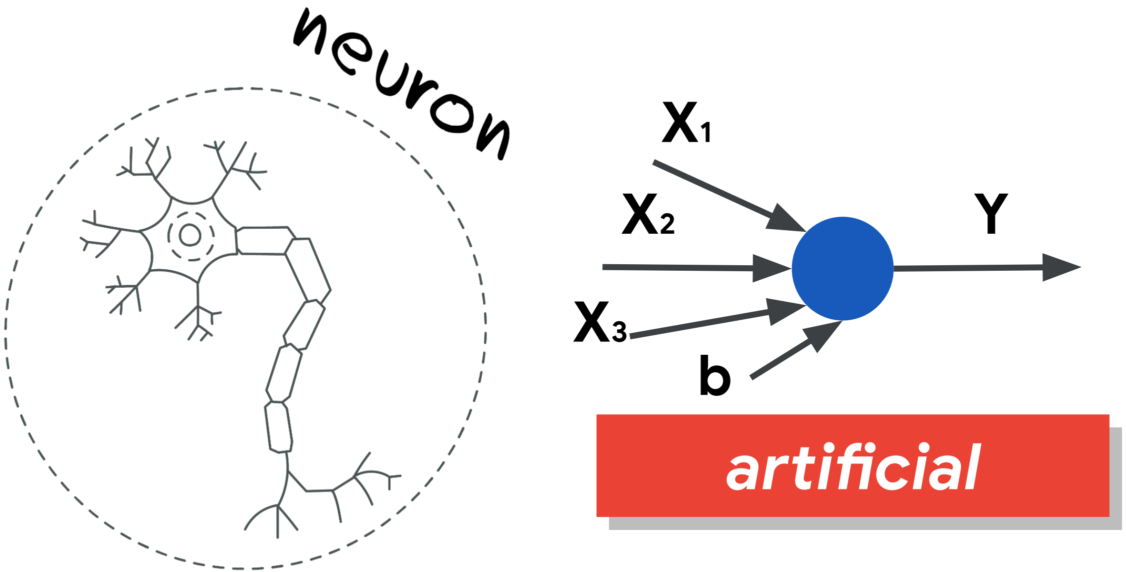
What is a **neural network**?



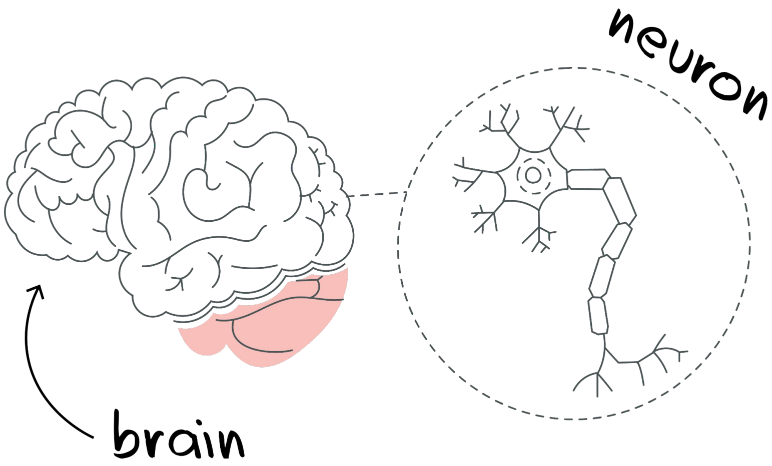
What is a **neural network**?

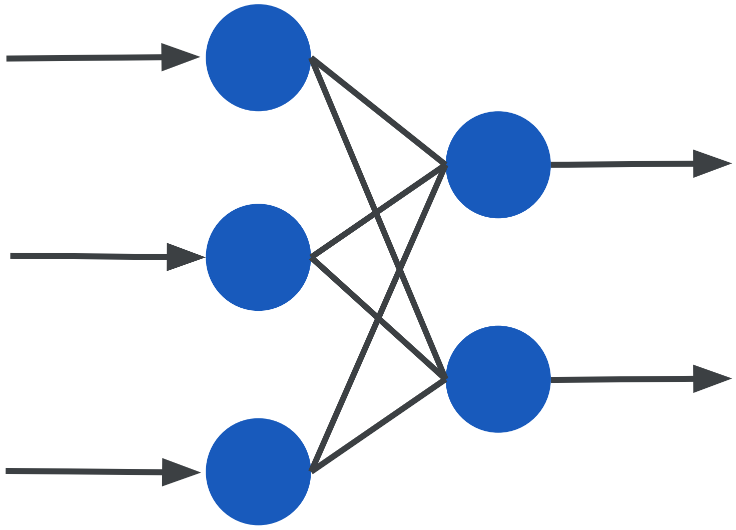


What is a **neural network**?



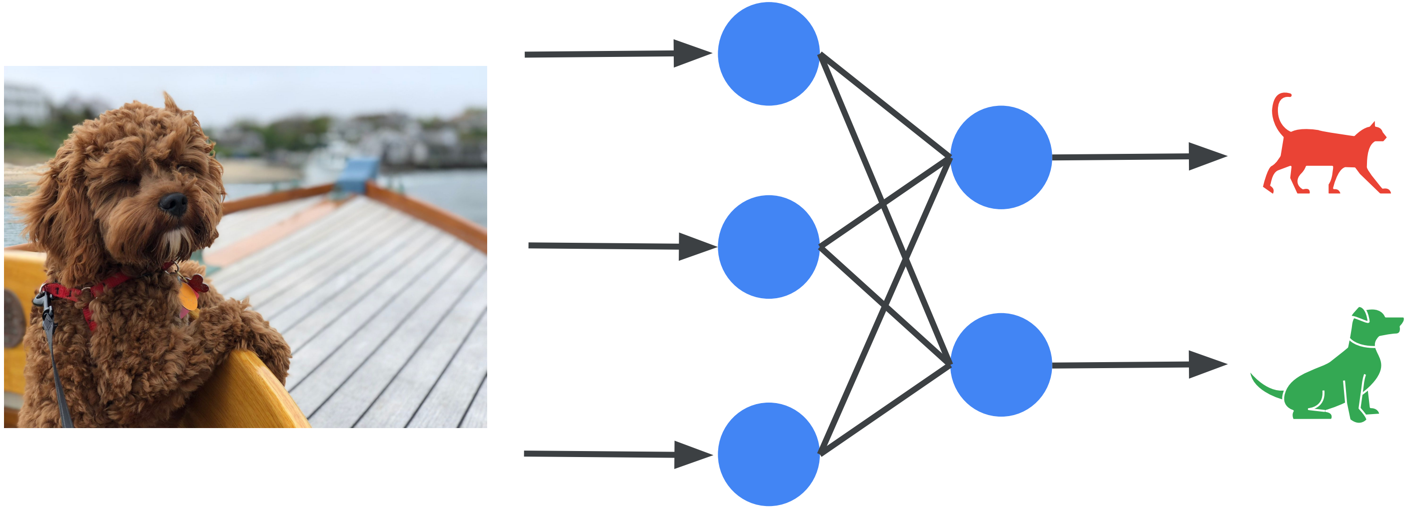
What is a **neural network**?



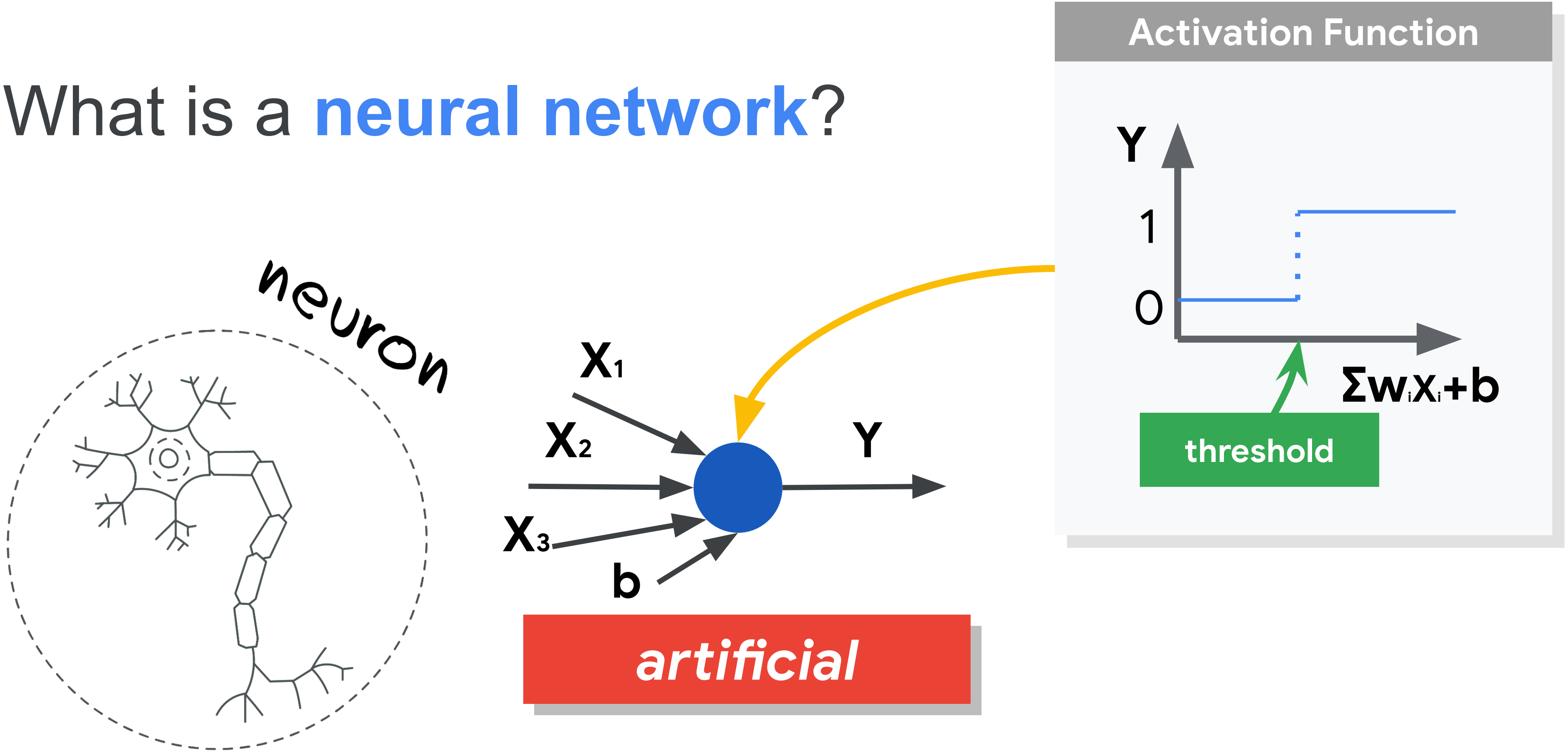


artificial

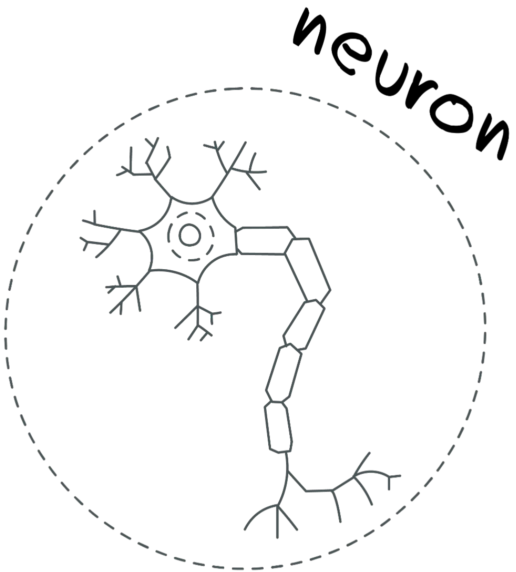
What is a **neural network**?

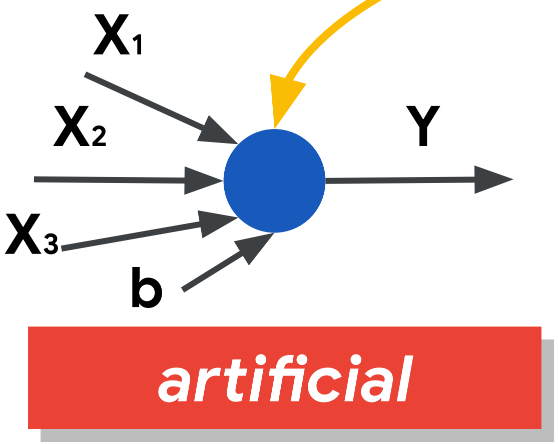


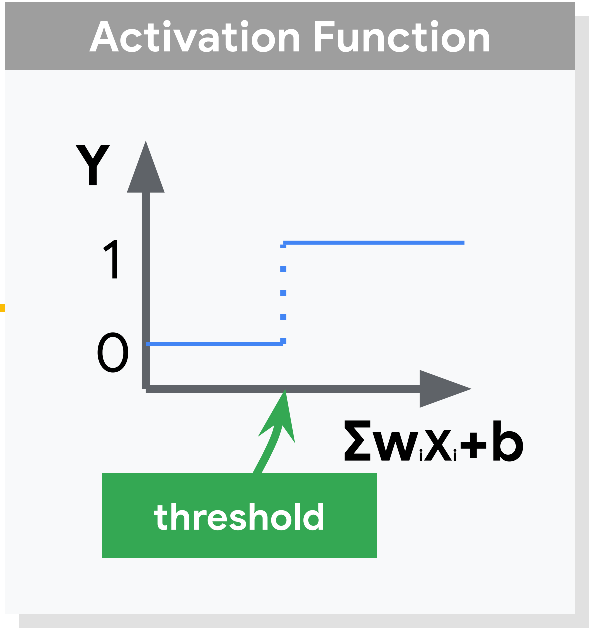
What is a **neural network**?



What is a **neural network**?



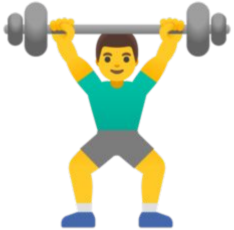




$Y = \Sigma w_i x_i + b$

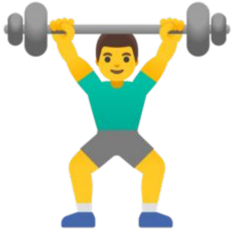
So training the model is finding the right values for w_i and b

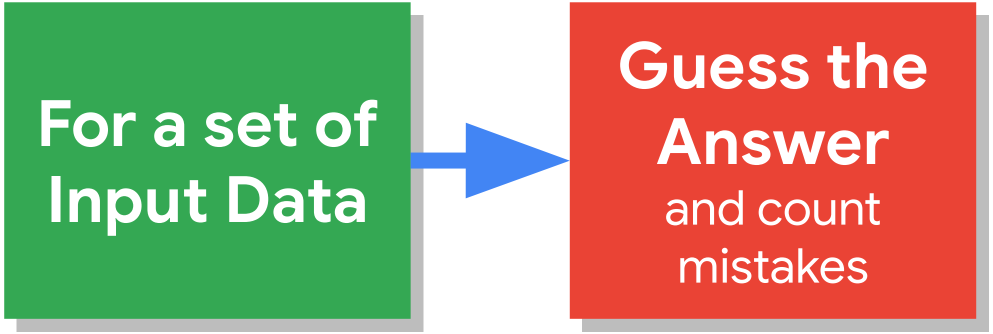
Training the machine



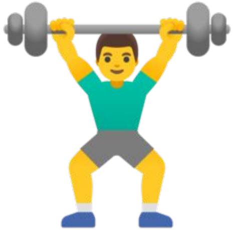
For a set of Input Data

Training the machine



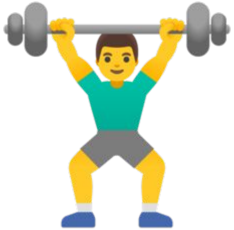


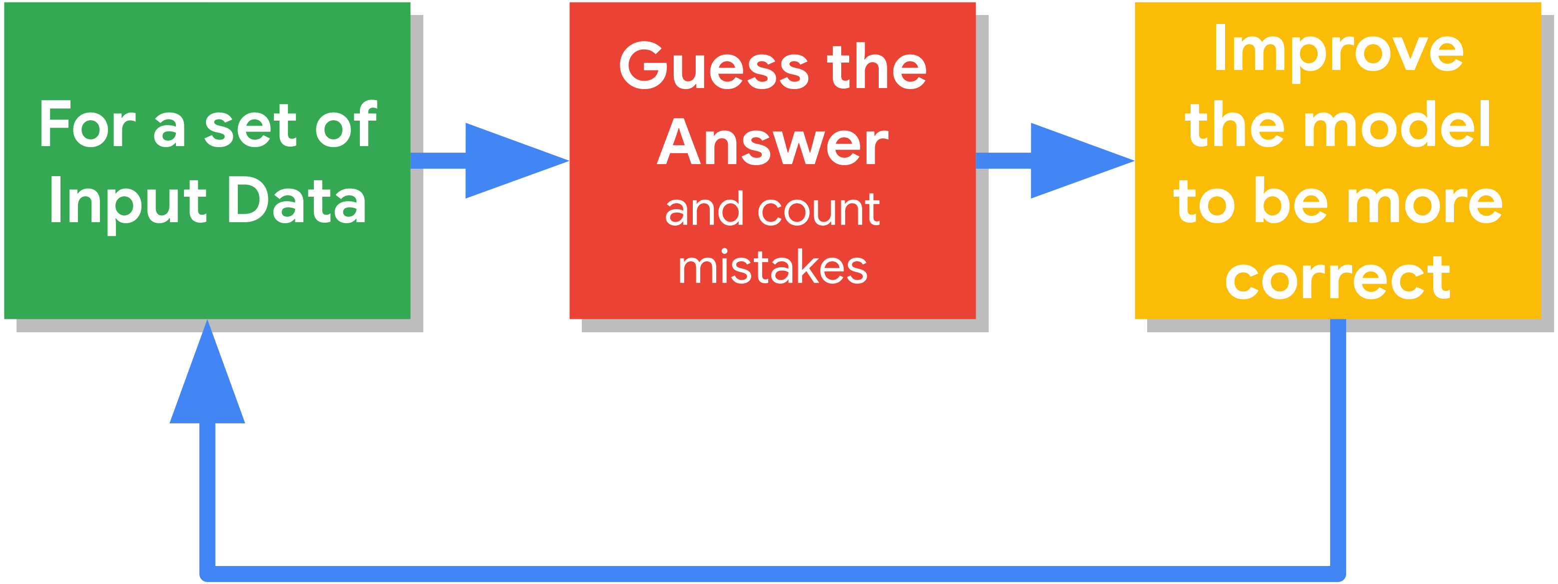
Training the machine



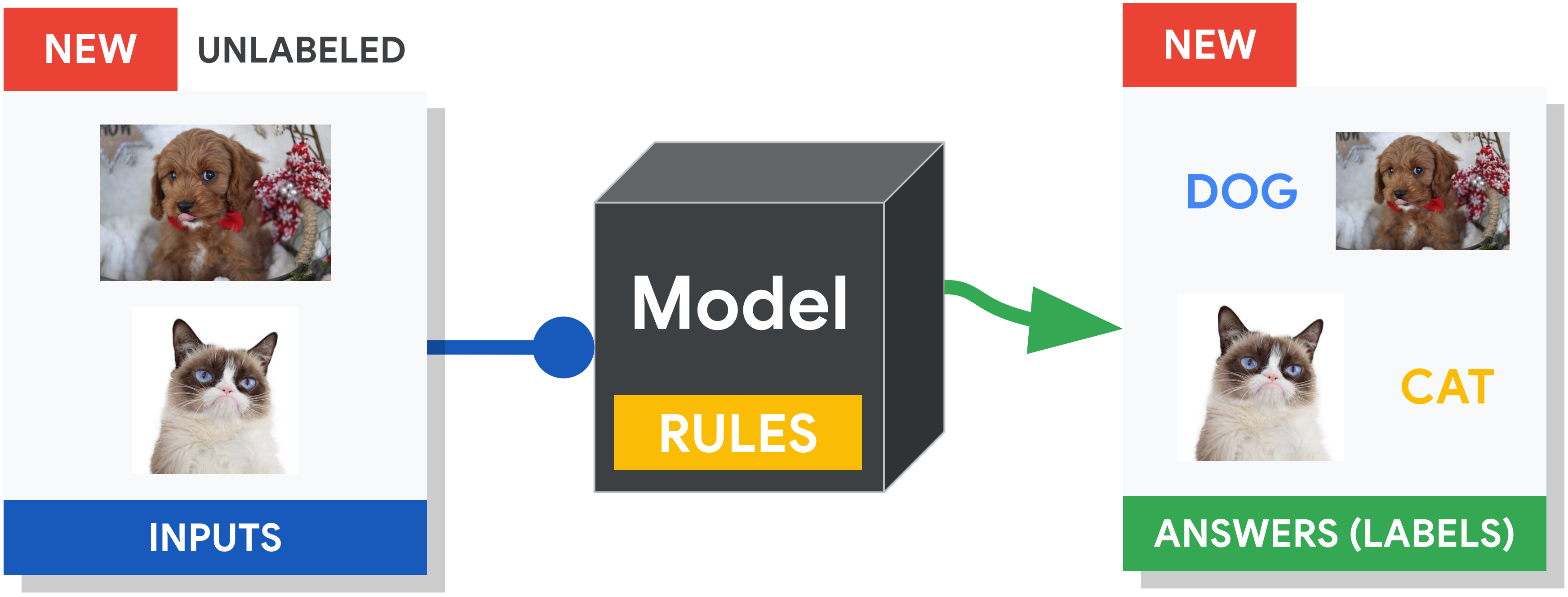
For a set of Input Data → **Guess the Answer** and count mistakes → **Improve the model to be more correct**

Training the machine



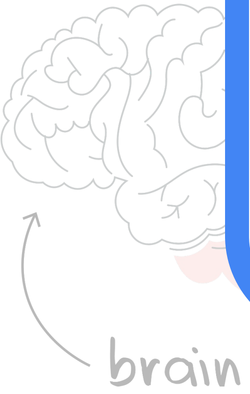


After it's **learned** use it for **inference**:



What is a **neural network**?

To learn more about the **math behind neural network training** there is a nice series of videos here:
https://www.youtube.com/playlist?list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi



artificial

Today's Agenda

- Deep ML Background
  - How does (Deep) Machine Learning Work?
  - **Exploring Deep ML through Computer Vision**
- Hands-on Computer Vision: Thing Translator
- The Tiny Machine Learning Workflow
- Keyword Spotting (KWS) Data Collection
- KWS Preprocessing and Training
- Deployment Challenges and Opportunities for Embedded ML
- Summary

Computer Vision is Hard

What color are the pants and the shirt?

Slide Credit: Hamilton Chong

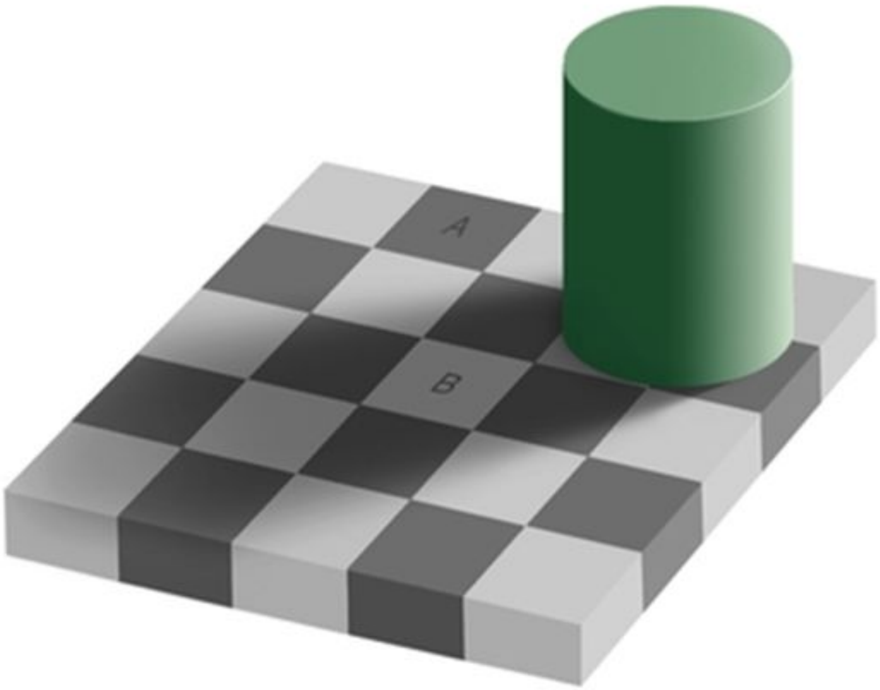
Computer Vision is Hard

Slide Credit: Hamilton Chong

Computer Vision is Hard

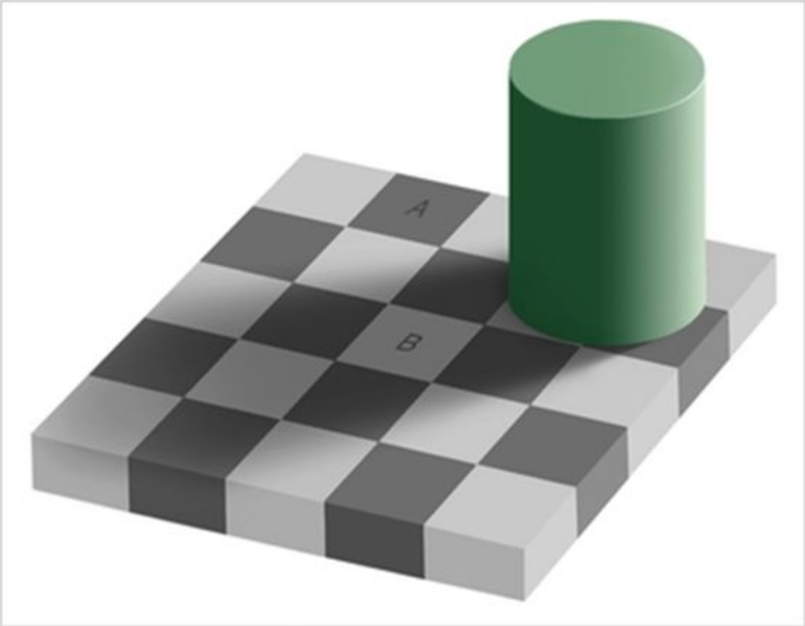
Slide Credit: Hamilton Chong

Computer Vision is Hard

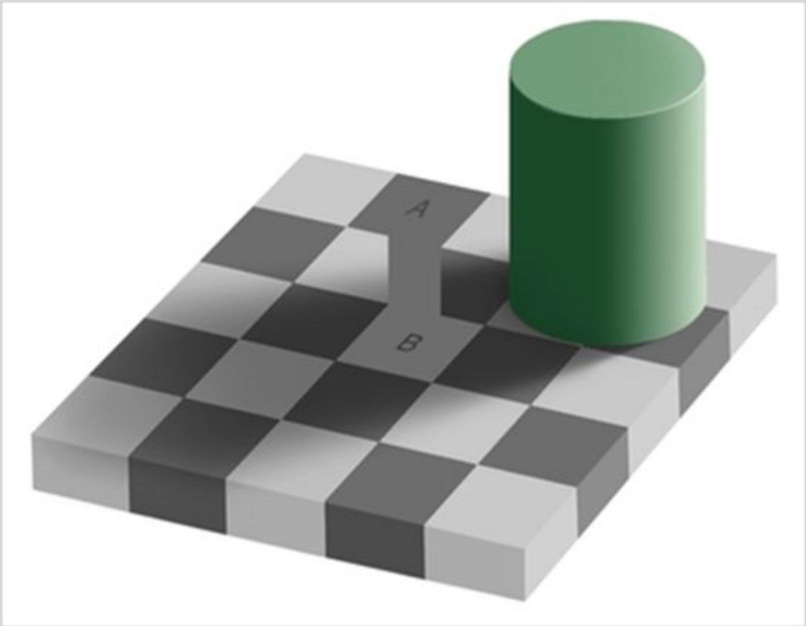


Is square A or B darker in color?

Computer Vision is Hard

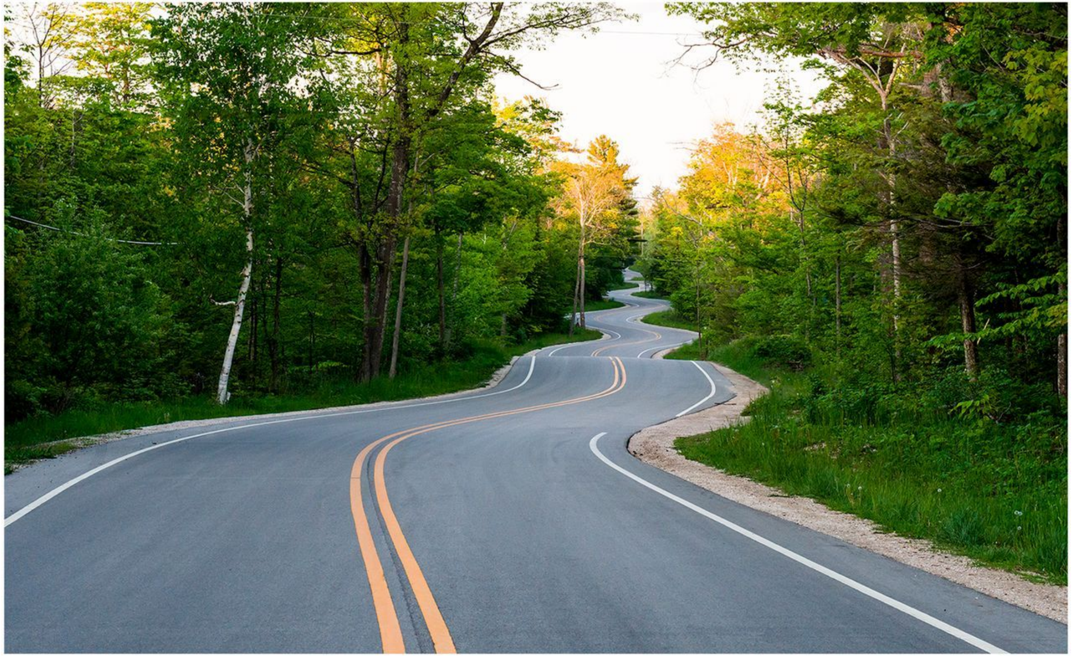


Areas of the image A and B are the same color



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?



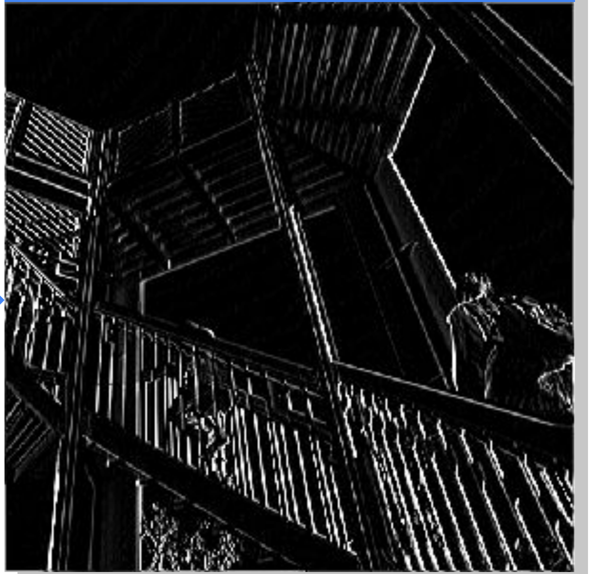
What **Features** of the image might be important for self driving cars?

Maybe straight lines to see the lanes of the road?

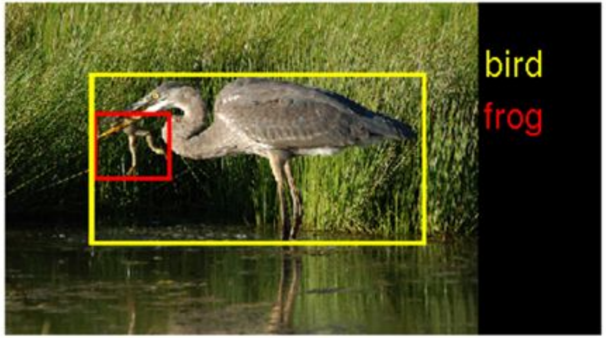
Features can be found with **Convolutions**

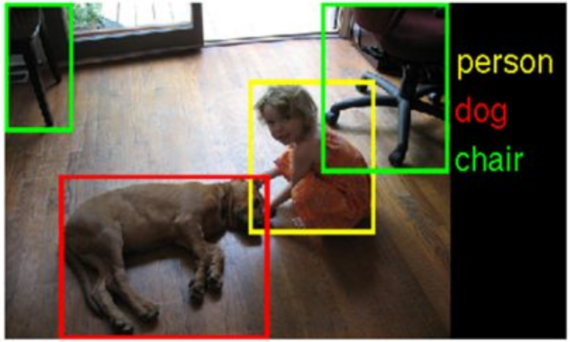
-1	0	1
-2	0	2
-1	0	1

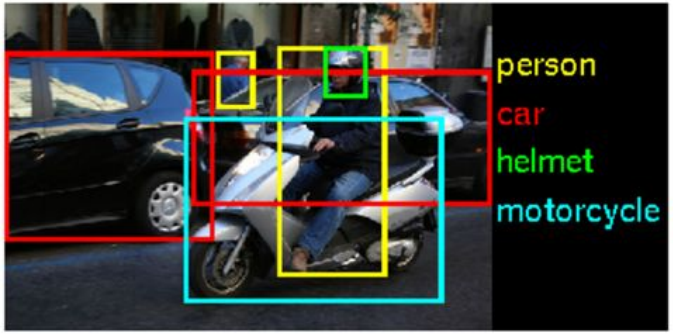
Features



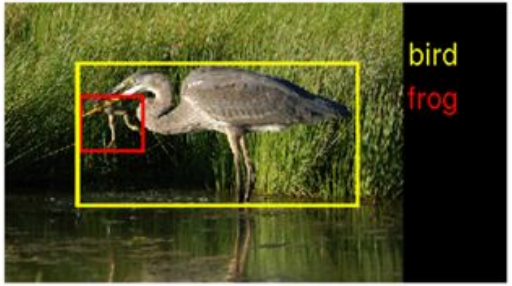
What features are needed for Object Detection?

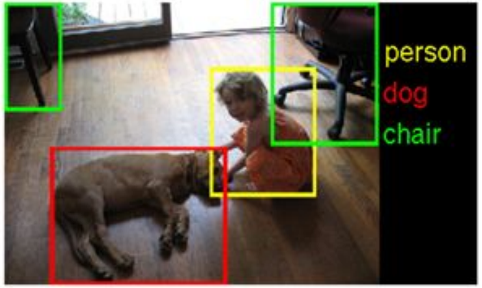


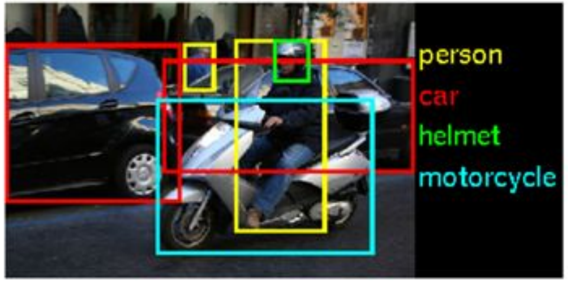




What features are needed for Object Detection?







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

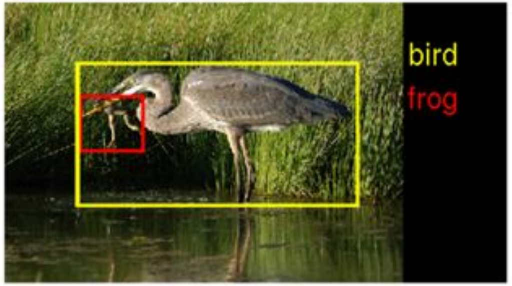
What features are needed for Object Detection?

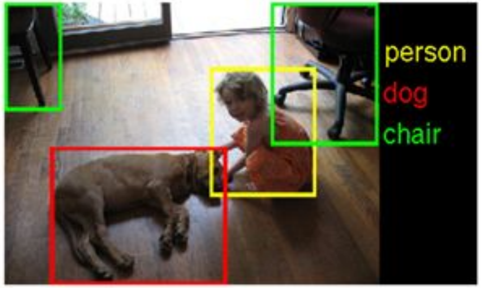
Traditional Machine Learning Flow

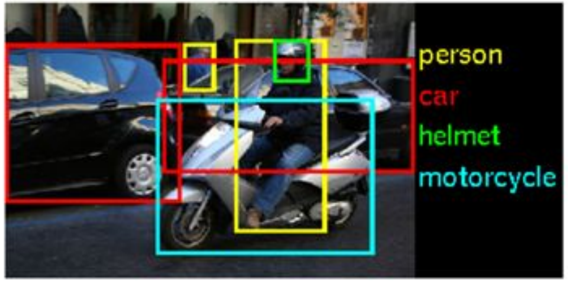
Regression, Clustering, etc.

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

What features are needed for Object Detection?



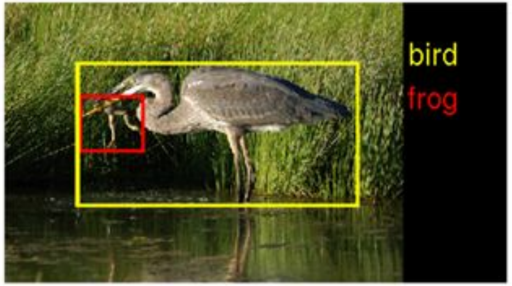


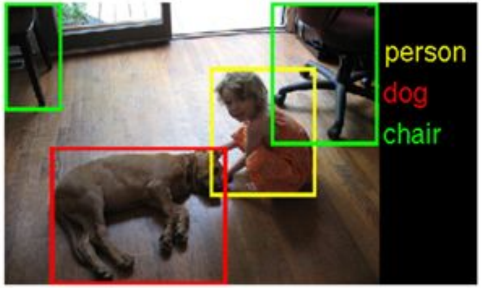


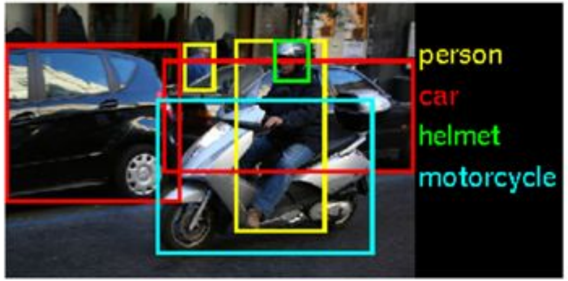
In 2010 teams had
75-50% error

In 2011 teams had
75-25% error

What features are needed for Object Detection?

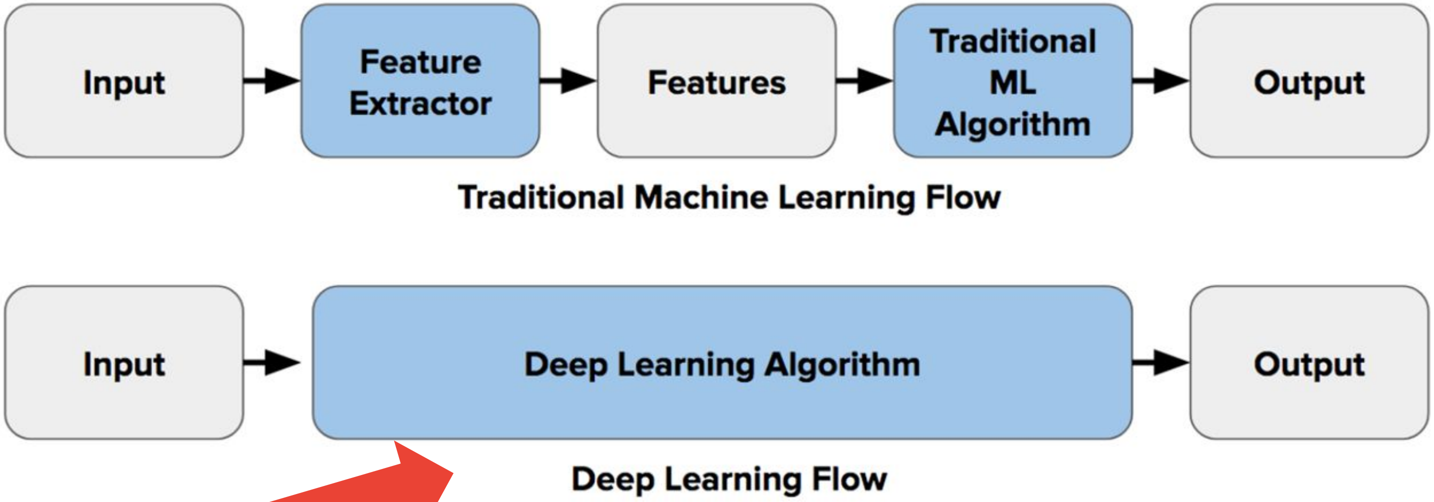






In 2012 still no team had less than 25% error barrier except **AlexNet at 15%**

What features are needed for Object Detection?

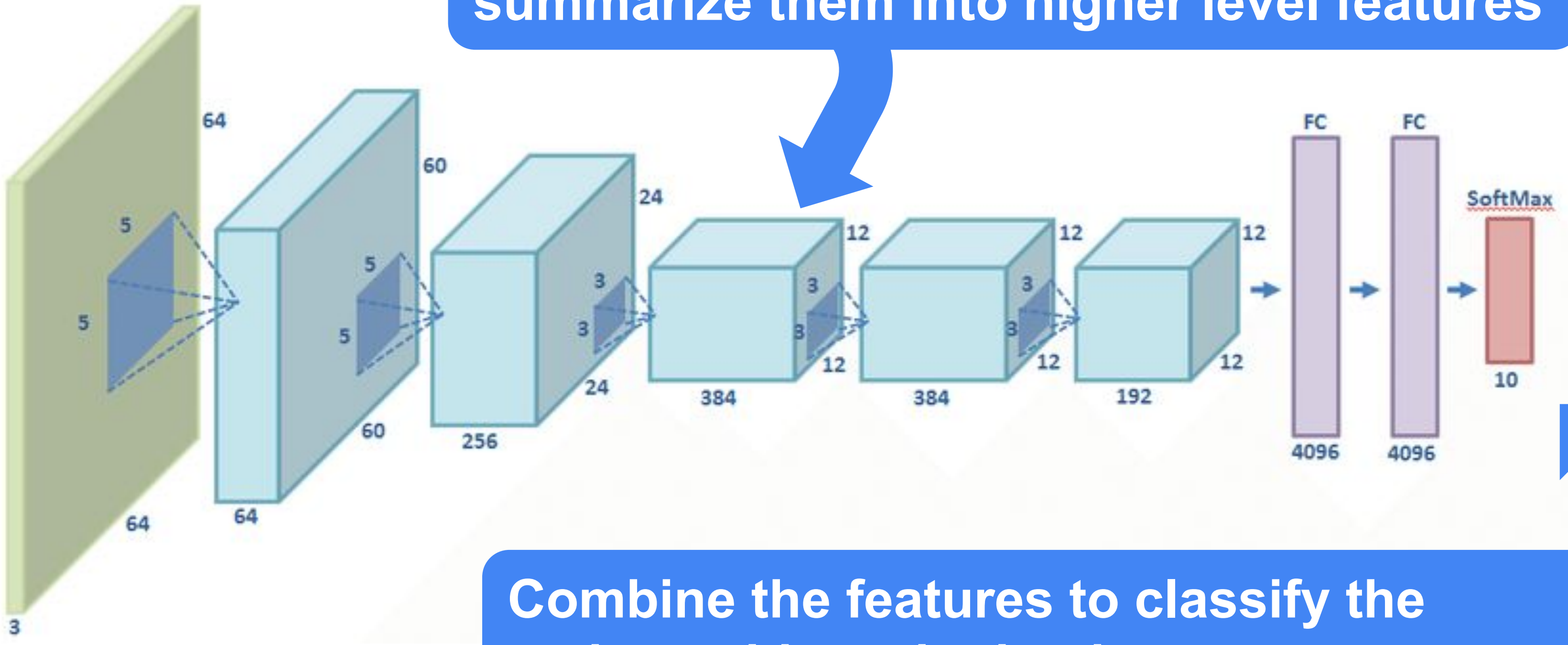


Let the computer figure out its own features and how to combine them!

[AlexNet Paper](#)

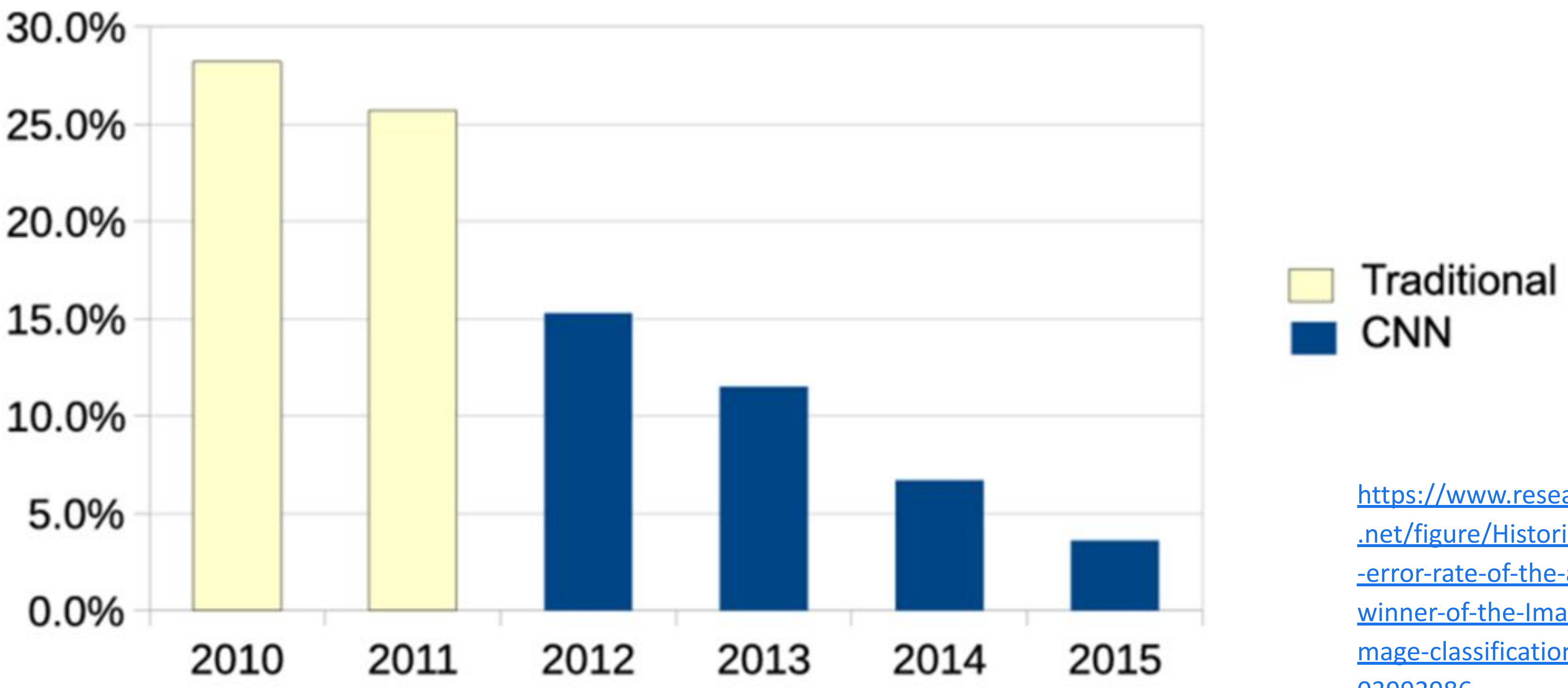
AlexNet

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



Combine the features to classify the various objects in the dataset

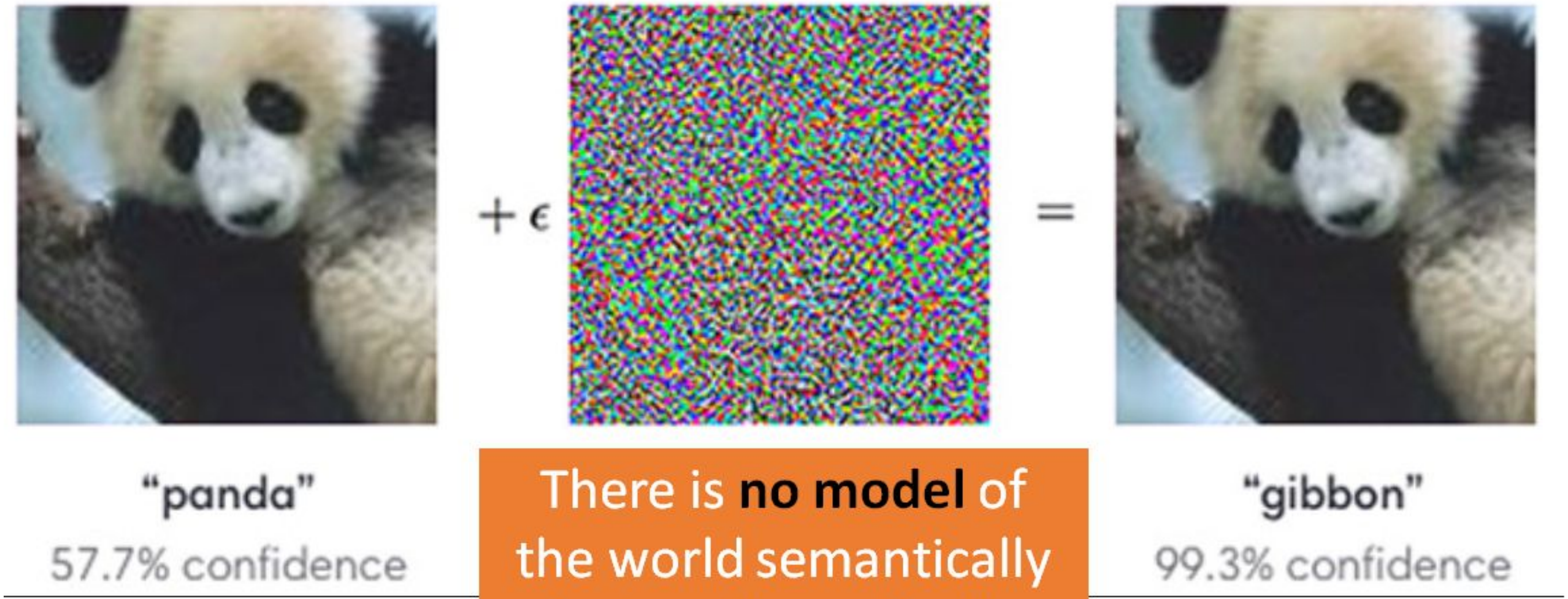
What features are needed for Object Detection?



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

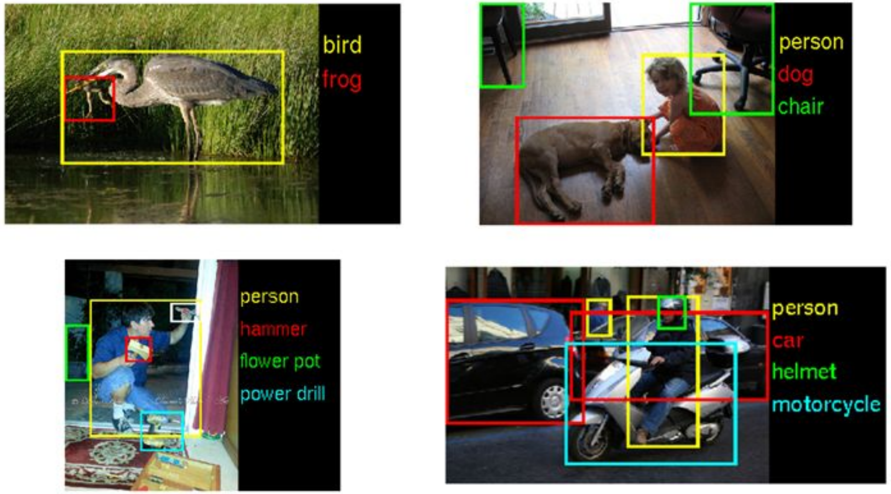
Today's Agenda

- Deep ML Background
- **Hands-on Computer Vision: Thing Translator**
- The Tiny Machine Learning Workflow
- Keyword Spotting (KWS) Data Collection
- KWS Preprocessing and Training
- Deployment Challenges and Opportunities for Embedded ML
- Summary

The **Thing Translator**

Open On Your Phone

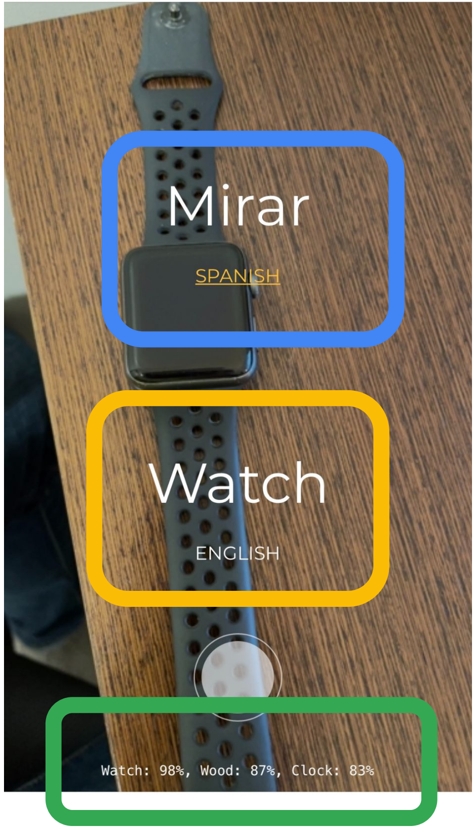
https://thing-translator.appspot.com/



The **Thing Translator**

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

Open On Your Phone



Today’s Agenda

- Deep ML Background
- Hands-on Computer Vision: Thing Translator
- **The Tiny Machine Learning Workflow**
- Keyword Spotting (KWS) Data Collection
- KWS Preprocessing and Training
- Deployment Challenges and Opportunities for Embedded ML
- Summary

What is Embedded Machine Learning (TinyML)?

TinyML

Fastest-growing field of **ML**

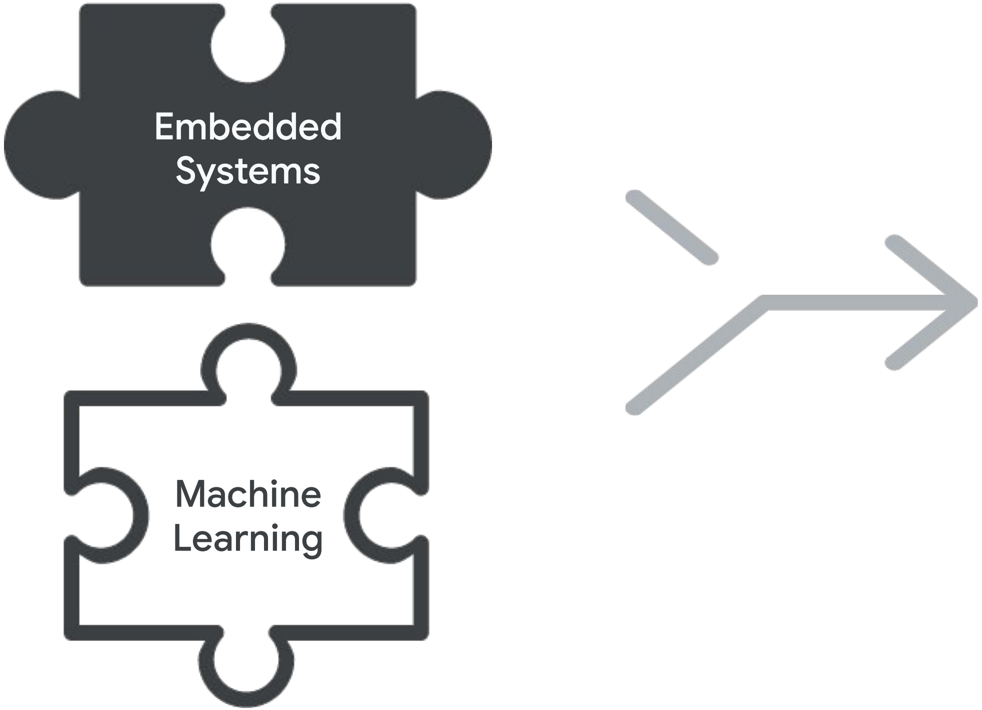
Algorithms, hardware, software

On-device sensor analytics

Low power consumption

Always-on ML

Battery-operated


Embedded
Systems
Machine
Learning

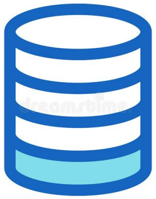
TinyML


TINY MACHINE
LEARNING KIT
ARDUINO
edX
TensorFlow

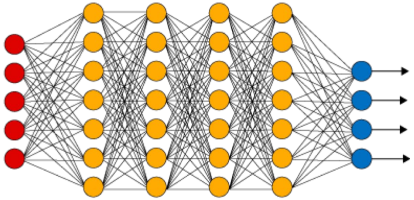
The TinyML Workflow

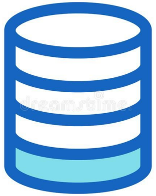
Collect Data

Preprocess Data

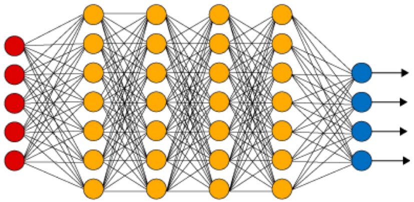


The TinyML Workflow

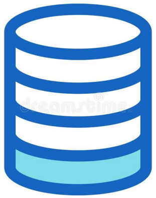




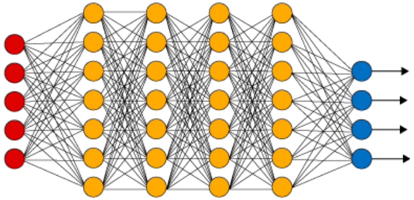
The TinyML Workflow

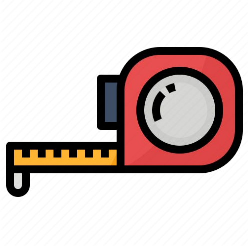


Collect Data → Preprocess Data → Design a Model → Train a Model → Evaluate Optimize → Convert Model

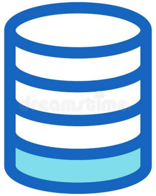


The TinyML Workflow

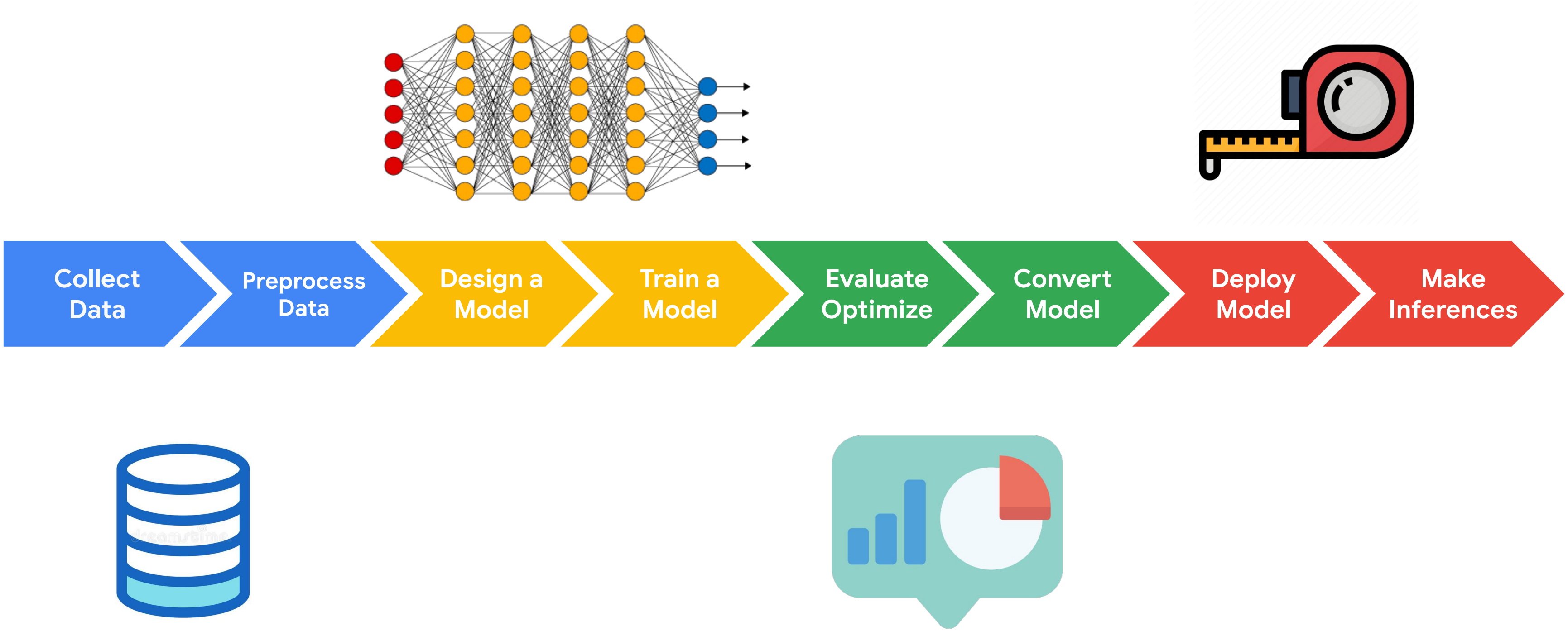




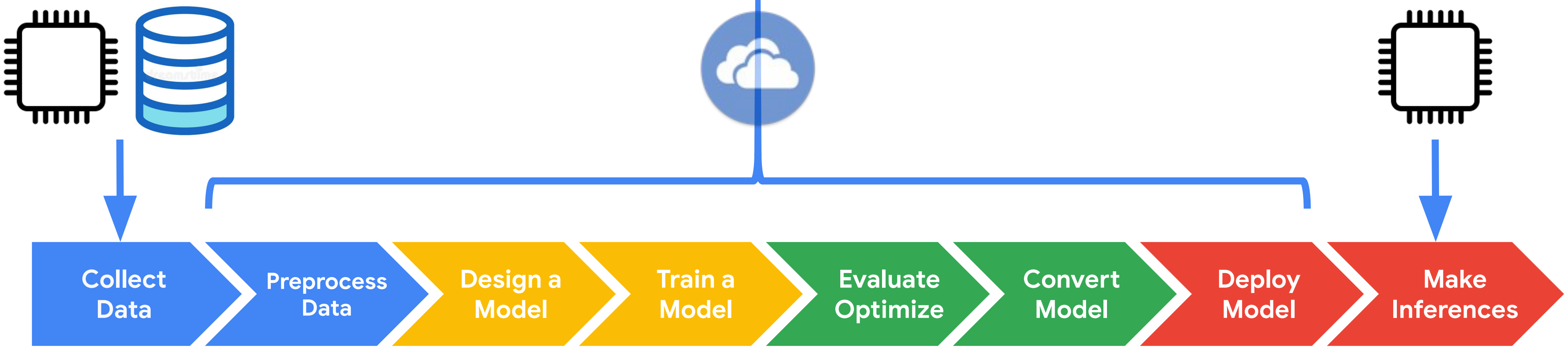
Collect Data → Preprocess Data → Design a Model → Train a Model → Evaluate Optimize → Convert Model → Deploy Model → Make Inferences



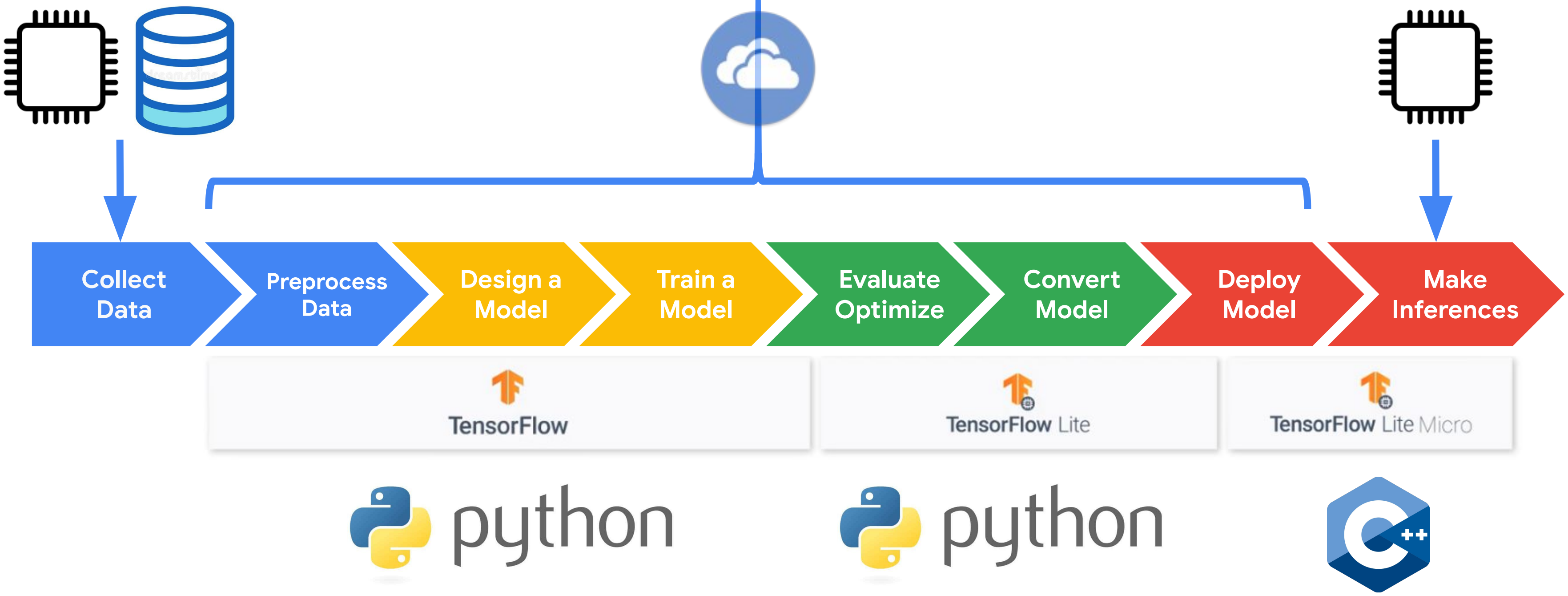
The TinyML Workflow ("What")



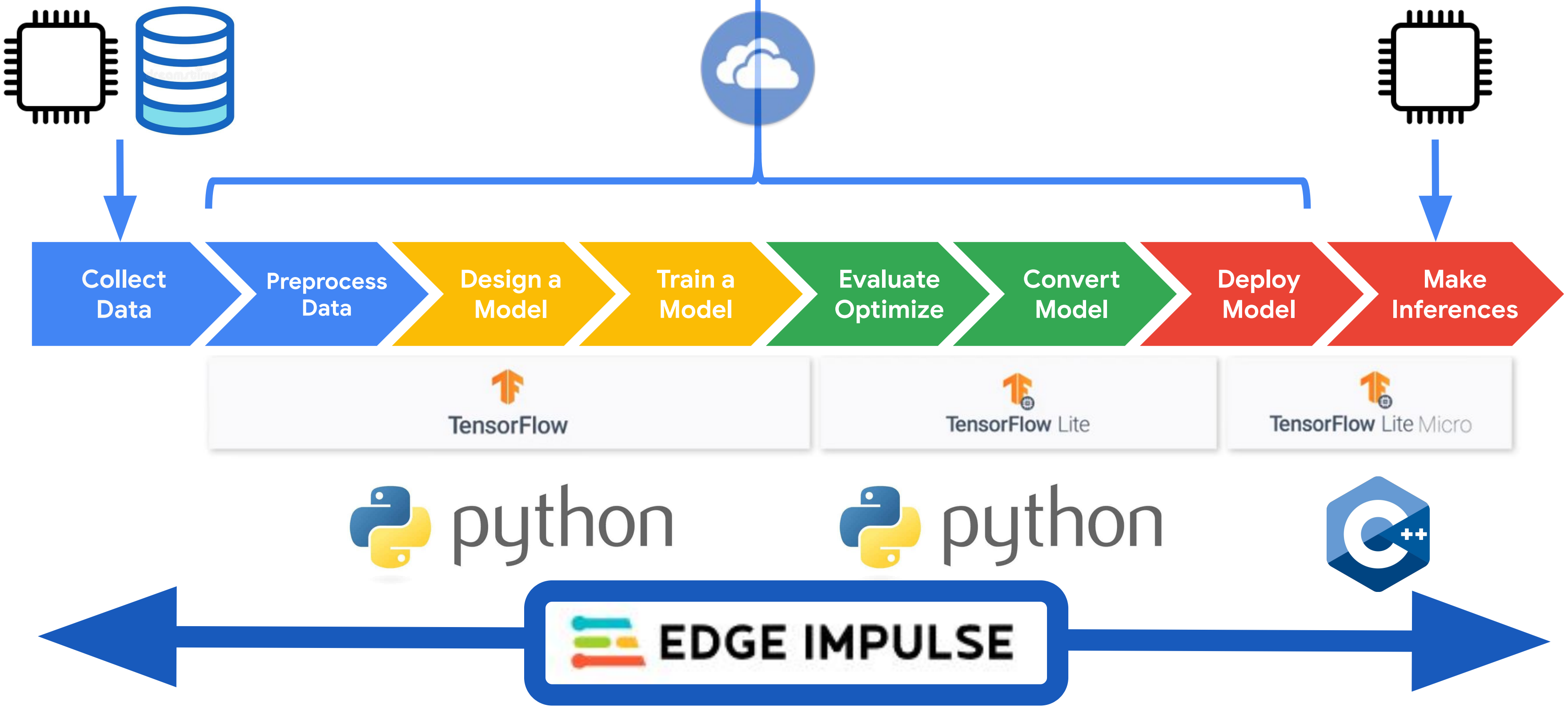
The TinyML Workflow ("Where")



The TinyML Workflow (“How”)



The TinyML Workflow ("How")

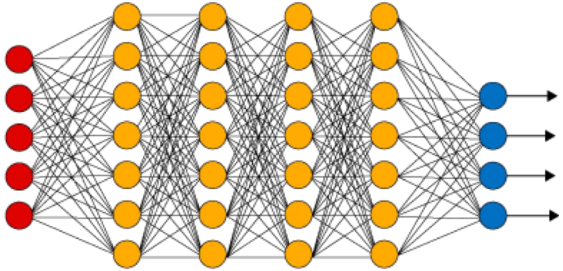


Today’s Agenda

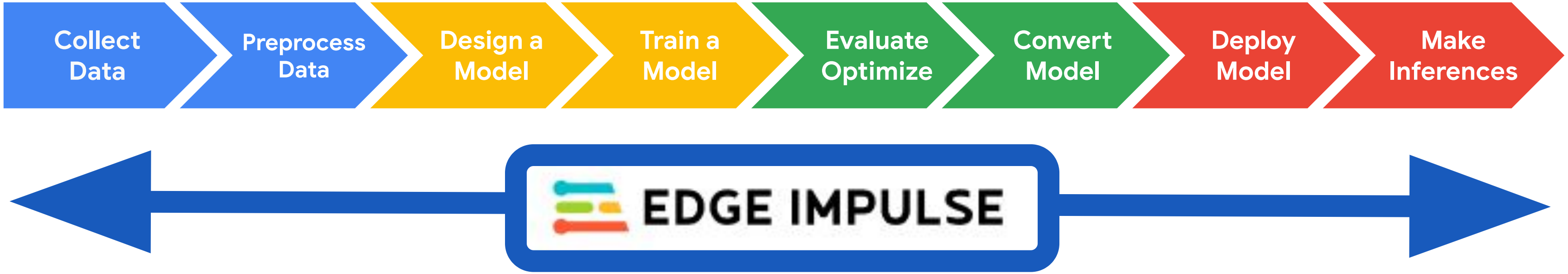
- Deep ML Background
- Hands-on Computer Vision: Thing Translator
- The Tiny Machine Learning Workflow
- **Keyword Spotting (KWS)** Data Collection
    - A Quick Primer on Data Engineering
    - Hands-on KWS Data Collection with Edge Impulse
- KWS Preprocessing and Training
- Deployment Challenges and Opportunities for Embedded ML
- Summary

Keyword Spotting in One Slide

If we **pick a simple task** to only identifying a **few key words** we can then use a **small model** and train it with **little data** and fit it onto an **embedded device**



The **TinyML** Workflow



Today's Agenda

- Deep ML Background
- Hands-on Computer Vision: Thing Translator
- The Tiny Machine Learning Workflow
- Keyword Spotting (KWS) Data Collection
  - **A Quick Primer on Data Engineering**
  - Hands-on KWS Data Collection with Edge Impulse
- KWS Preprocessing and Training
- Deployment Challenges and Opportunities for Embedded ML
- Summary

Data Engineering for KWS (Part 1)

(How to collect good data)

Data Engineering for KWS (Part 1)

(How to collect good data)

Who will use your ML model?

- What **languages** will they speak?
- What **accents** will they have?
- Will they use **slang** or formal diction?

Data Engineering for KWS (Part 1)

(How to collect good data)

Who will use your ML model?

- What **languages** will they speak?
- What **accents** will they have?
- Will they use **slang** or formal diction?

Where will your ML model be used?

- Will there be **background noise**?
- **How far** will users be from the microphone?
- Will there be **echos**?

Data Engineering for KWS (Part 1)
(How to collect good data)

Who will use your ML model?

- What **languages** will they speak?
- What **accents** will they have?
- Will they use **slang** or formal diction?

Where will your ML model be used?

- Will there be **background noise**?
- **How far** will users be from the microphone?
- Will there be **echos**?

Why will your ML model be used? **Why** those Keywords?

- What **tone of voice** will be used?
- Are your **keywords commonly** used? (aka will you get a lot of false positives)
- What about false negatives?

Data Engineering for KWS (Part 1)

(How to collect good data)

There are a lot more things to consider to **eliminate bias** and **protect privacy** when collecting data that we will talk about in future sessions!

ML model be used?

Why those Keywords?

...aka will you get a lot of false positives)

- What about false negatives?

Tips and Tricks for Custom KWS

- Pick **uncommon words** for Keywords
- Record lots of “**other words**”
- Record in the **location** you are going to be **deploying**
- Get **your end users** to help you build a dataset
- Record with the same **hardware** you will **deploy**
- Always **test** and then **improve** your dataset and model

Tips and Tricks for Custom KWS

Today we are just working on a demo so to give our demo the the best chance of working we will:

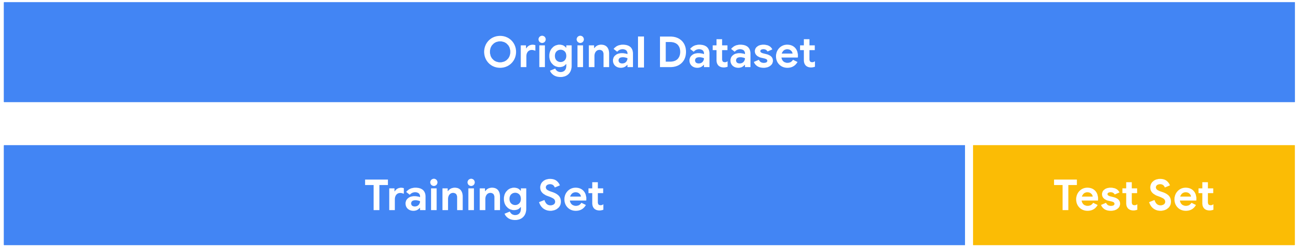
1. Stay in one spot
2. Only record ourselves
3. Use common words (yes, no)
4. Only test ourselves

(we're cheating)

Data Engineering for KWS (Part 2)

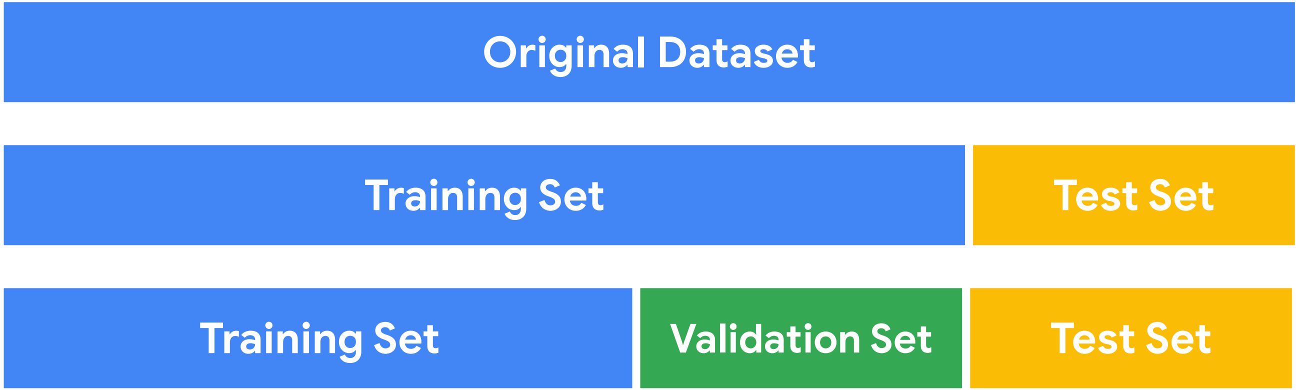
(how to test with our data)

Data Engineering for KWS (Part 2)
(how to test with our data)

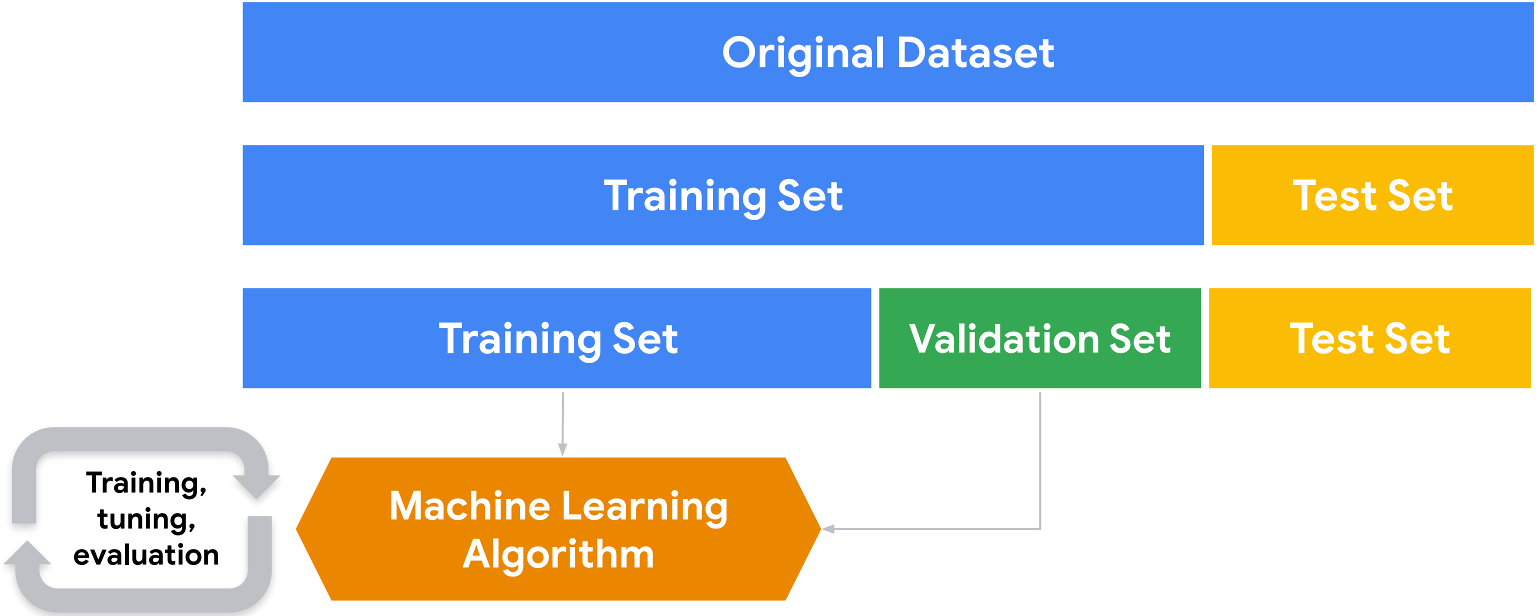


Data Engineering for KWS (Part 2)

(how to test with our data)

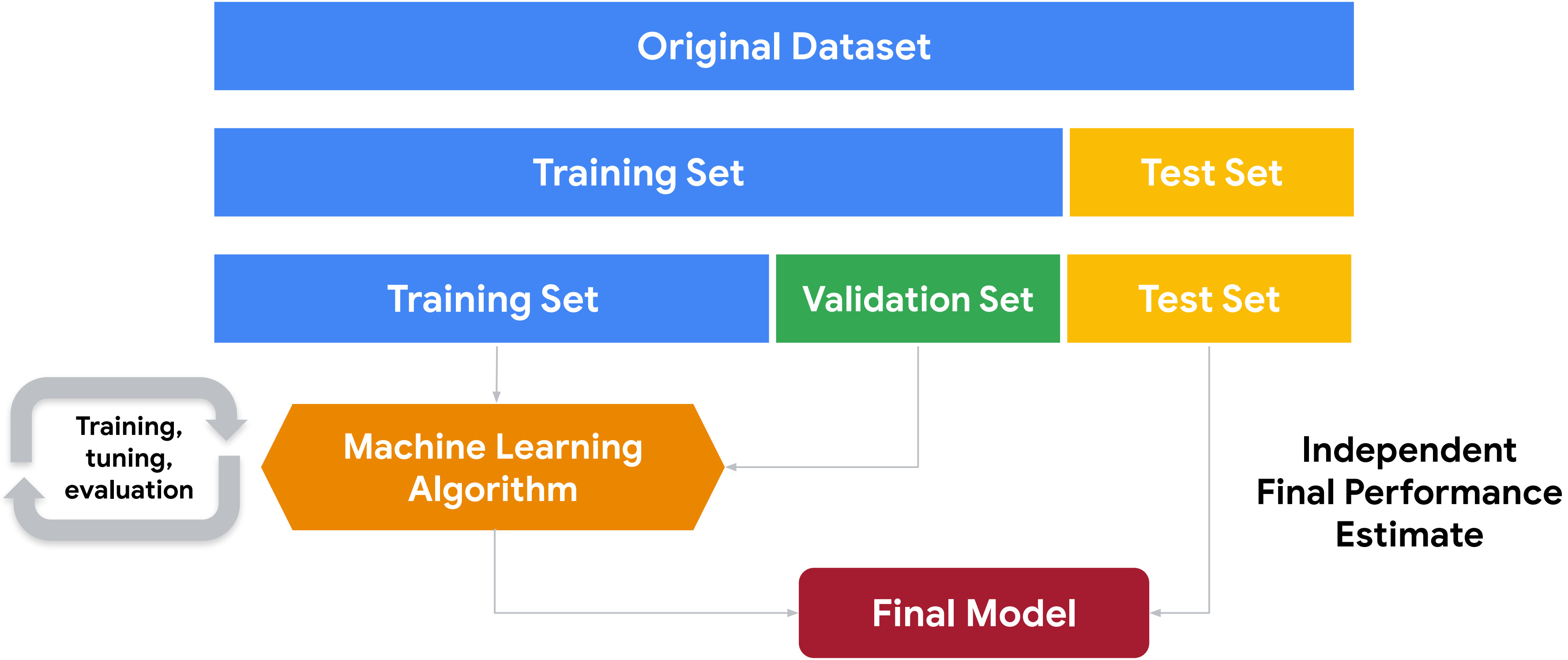


Data Engineering for KWS (Part 2)
(how to test with our data)



Data Engineering for KWS (Part 2)

(how to test with our data)



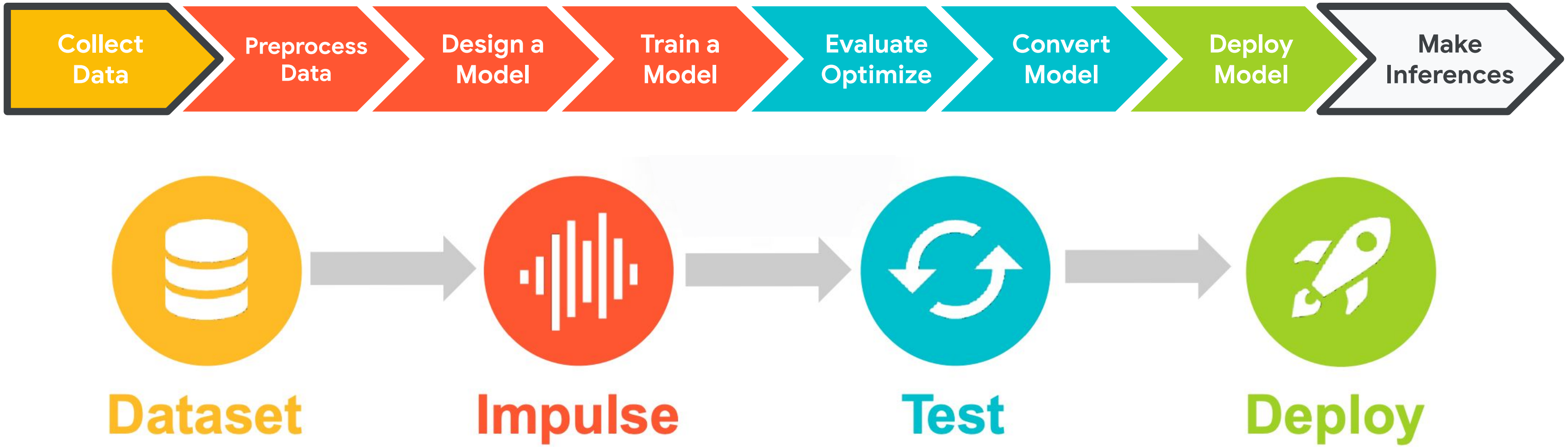
Today's Agenda

- Deep ML Background
- Hands-on Computer Vision: Thing Translator
- The Tiny Machine Learning Workflow
- Keyword Spotting (KWS) Data Collection
    - A Quick Primer on Data Engineering
    - **Hands-on KWS Data Collection with Edge Impulse**
- KWS Preprocessing and Training
- Deployment Challenges and Opportunities for Embedded ML
- Summary

The TinyML Workflow using Edge Impulse

Today we'll also collect all of our data using Edge Impulse...

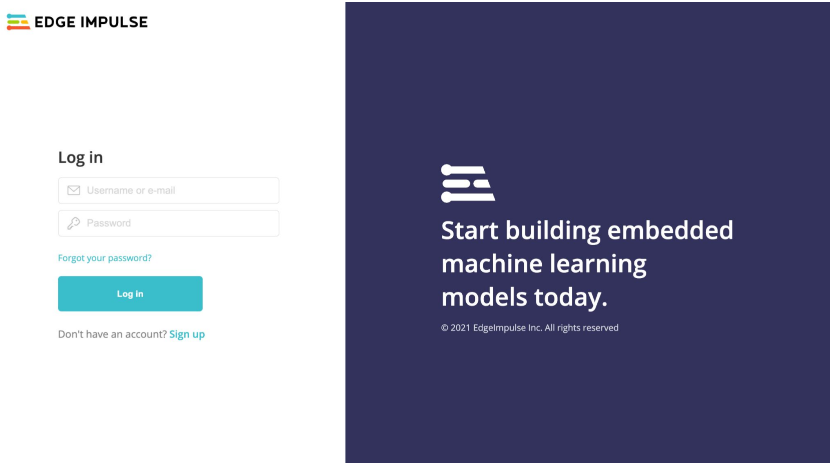
...and deploy to your cell phone as well

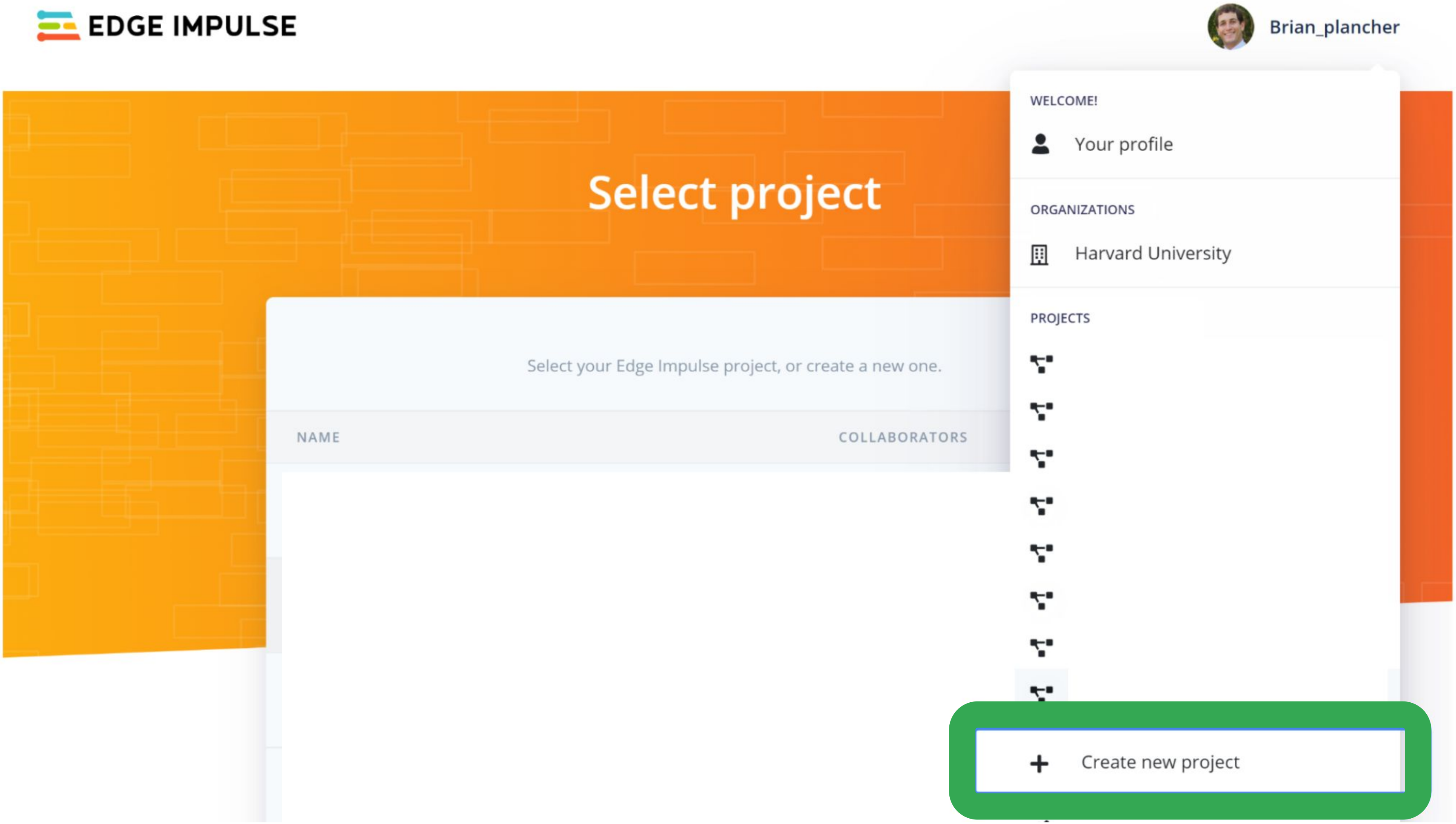


Create an Edge Impulse Account

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

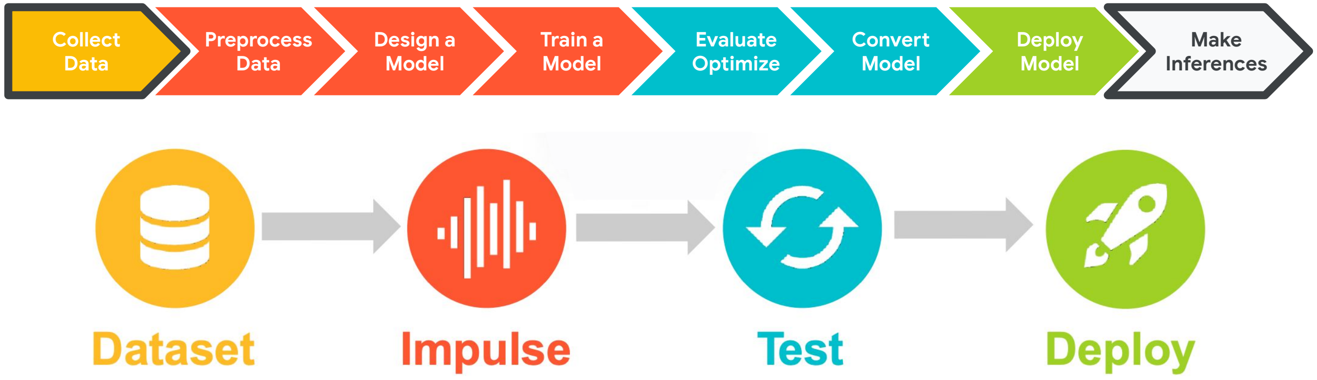
2. Validate your email by clicking the link in the email sent to your account's email address

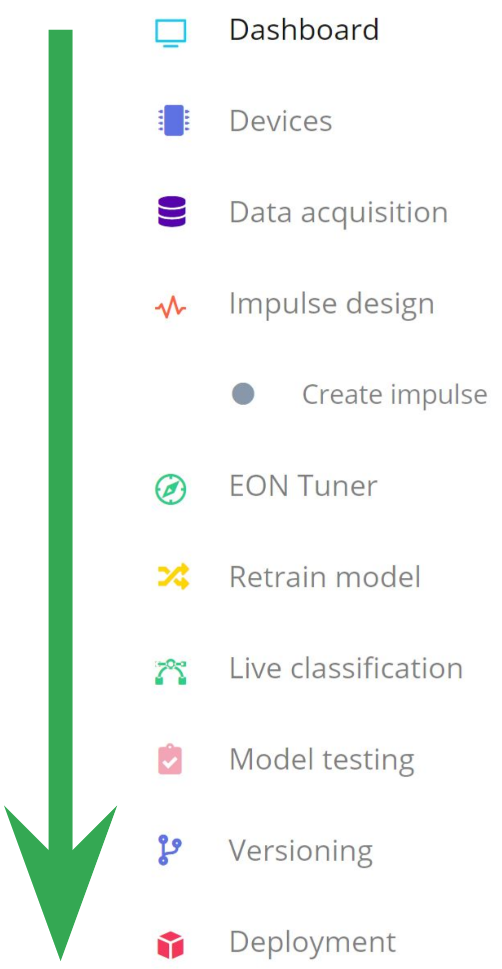




EDGE IMPULSE
Brian_plancher
Select project
Select your Edge Impulse project, or create a new one.
NAME
COLLABORATORS
WELCOME!
Your profile
ORGANIZATIONS
Harvard University
PROJECTS
Create new project

Edge Impulse Project Dashboard

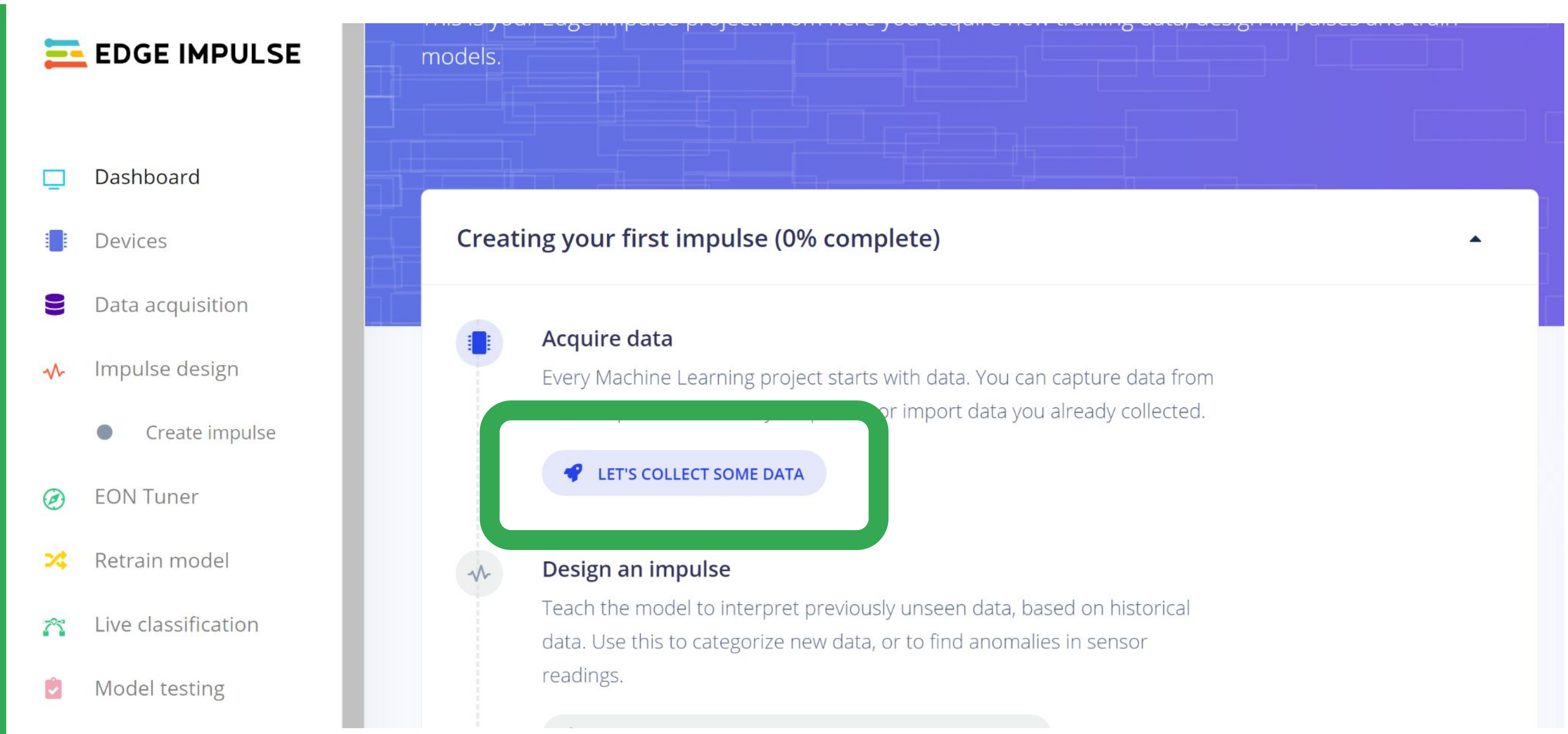


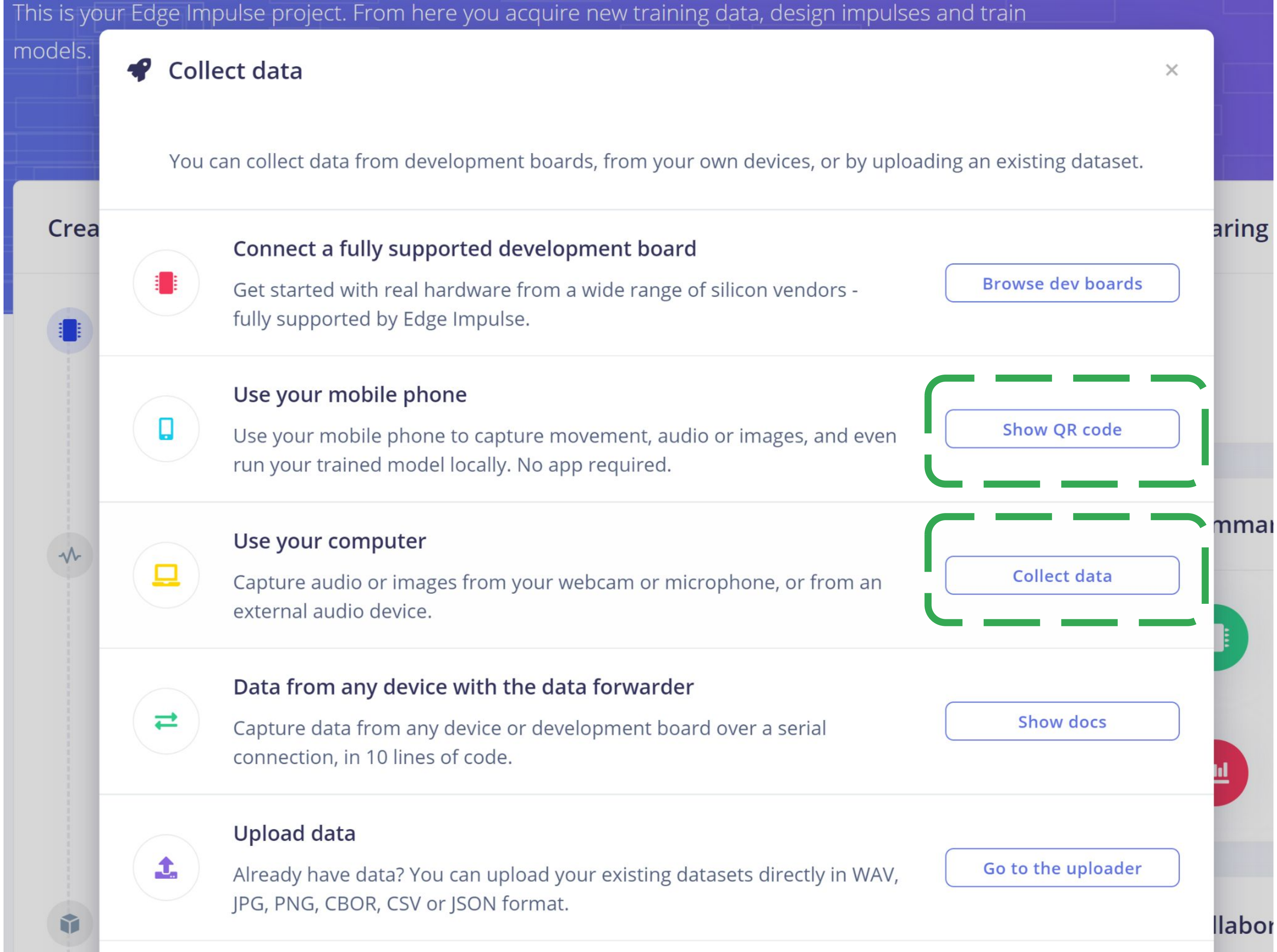


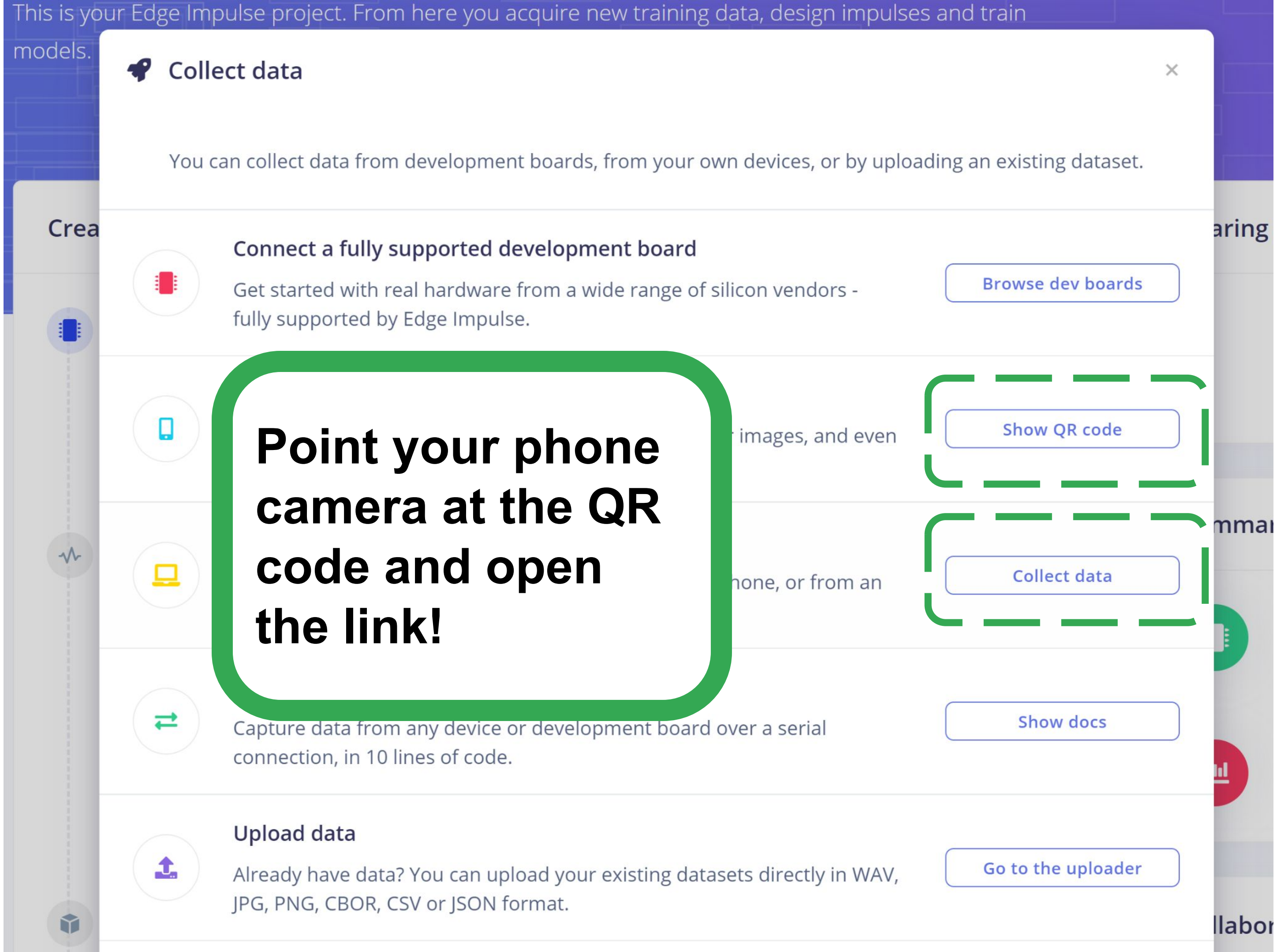
Activity: Create a Keyword Spotting Dataset

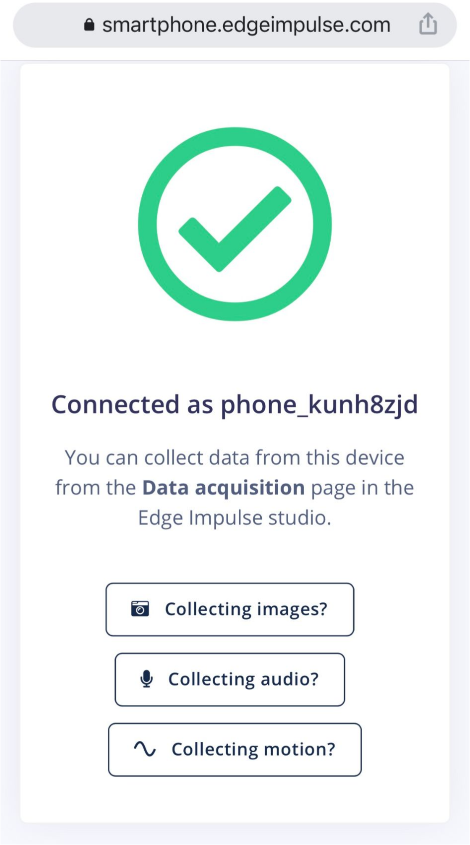
Collect **~30 samples each** of the following classes of data:

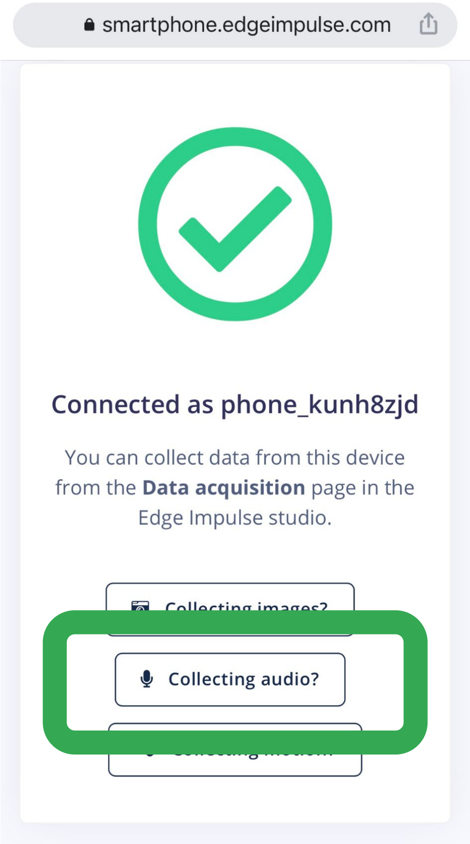
- **Keyword #1 “yes”** (label: yes) (length: 2 seconds)
- **Keyword #2 “no”** (label: no) (length: 2 seconds)
- **“Unknown” words** that are not the keyword **and background noise** (label: unknown) (length: 2 seconds)

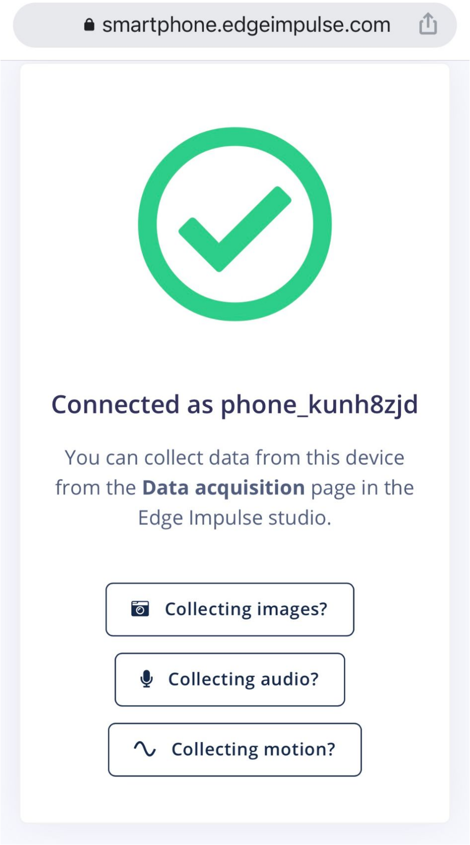

EDGE IMPULSE
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
EON Tuner
Retrain model
Live classification
Model testing
models.
Creating your first impulse (0% complete)
Acquire data
Every Machine Learning project starts with data. You can capture data from
or import data you already collected.
LET'S COLLECT SOME DATA
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.

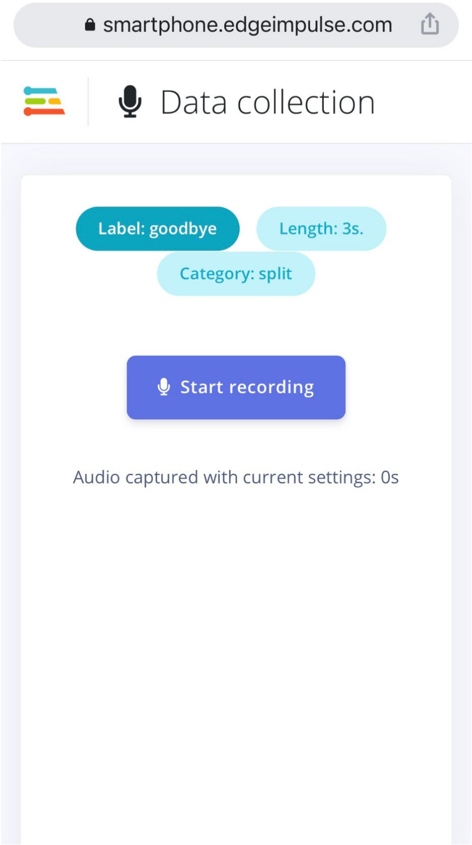

This is your Edge Impulse project. From here you acquire new training data, design impulses and train models.
Collect data
You can collect data from development boards, from your own devices, or by uploading an existing dataset.
Connect a fully supported development board
Get started with real hardware from a wide range of silicon vendors - fully supported by Edge Impulse.
Browse dev boards
Use your mobile phone
Use your mobile phone to capture movement, audio or images, and even run your trained model locally. No app required.
Show QR code
Use your computer
Capture audio or images from your webcam or microphone, or from an external audio device.
Collect data
Data from any device with the data forwarder
Capture data from any device or development board over a serial connection, in 10 lines of code.
Show docs
Upload data
Already have data? You can upload your existing datasets directly in WAV, JPG, PNG, CBOR, CSV or JSON format.
Go to the uploader

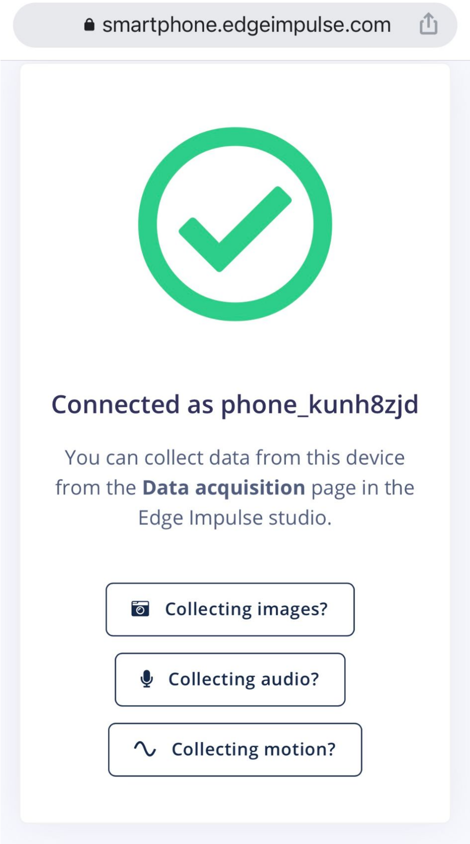

This is your Edge Impulse project. From here you acquire new training data, design impulses and train models.
Collect data
You can collect data from development boards, from your own devices, or by uploading an existing dataset.
Connect a fully supported development board
Get started with real hardware from a wide range of silicon vendors - fully supported by Edge Impulse.
Browse dev boards
images, and even
Show QR code
hone, or from an
Collect data
Capture data from any device or development board over a serial connection, in 10 lines of code.
Show docs
Upload data
Already have data? You can upload your existing datasets directly in WAV, JPG, PNG, CBOR, CSV or JSON format.
Go to the uploader
Point your phone camera at the QR code and open the link!

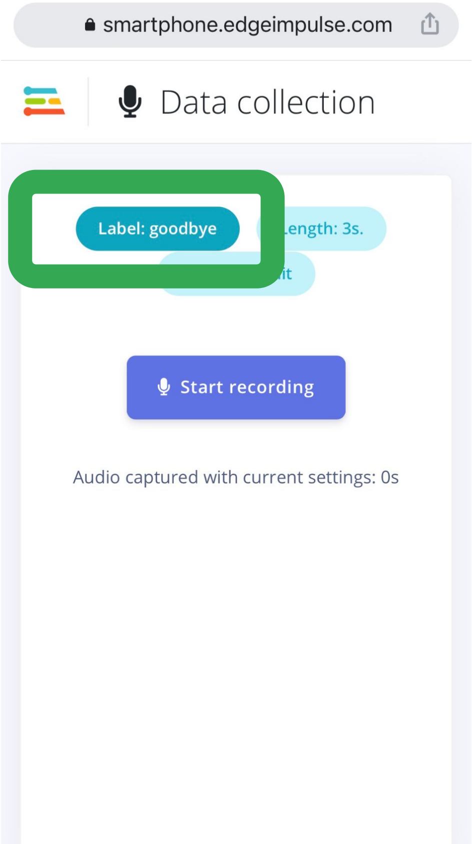

smartphone.edgeimpulse.com
Connected as phone_kunh8zjd
You can collect data from this device from the Data acquisition page in the Edge Impulse studio.
Collecting images?
Collecting audio?
Collecting motion?

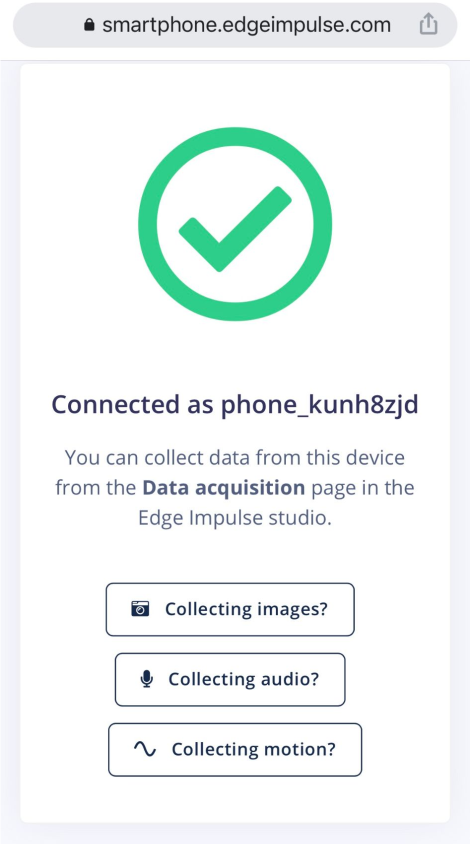

smartphone.edgeimpulse.com
Connected as phone_kunh8zjd
You can collect data from this device from the Data acquisition page in the Edge Impulse studio.
Collecting audio?

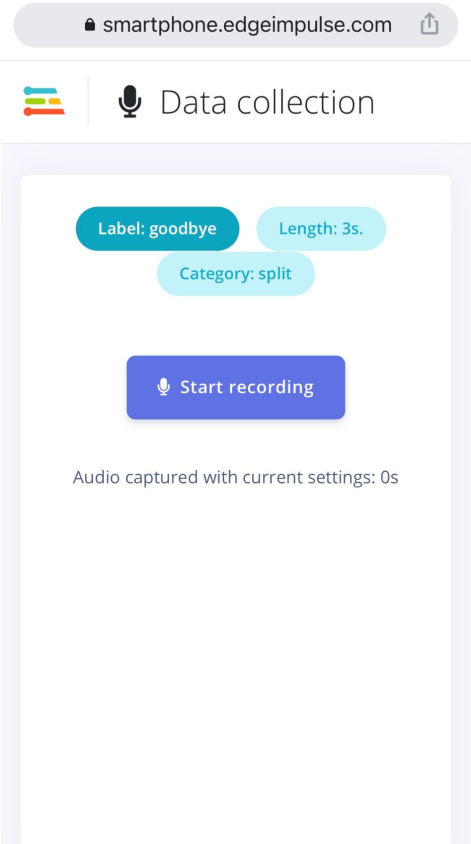

smartphone.edgeimpulse.com
Connected as phone_kunh8zjd
You can collect data from this device from the Data acquisition page in the Edge Impulse studio.
Collecting images?
Collecting audio?
Collecting motion?

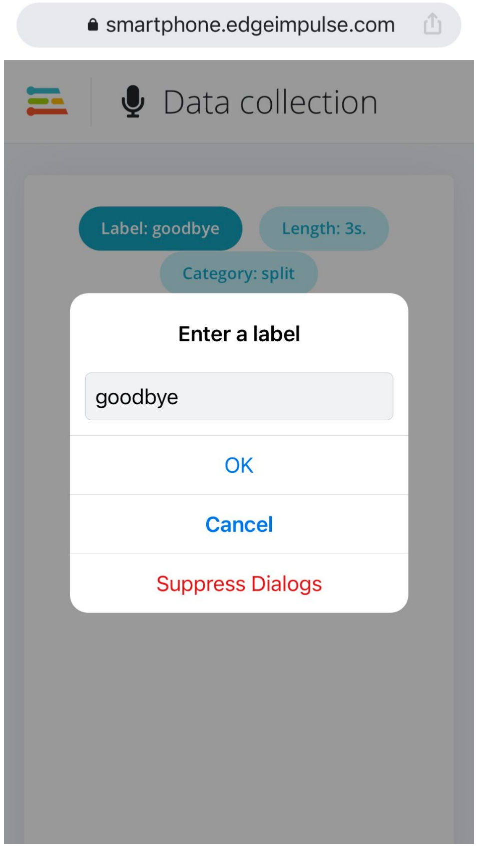

smartphone.edgeimpulse.com
Data collection
Label: goodbye
Length: 3s.
Category: split
Start recording
Audio captured with current settings: 0s

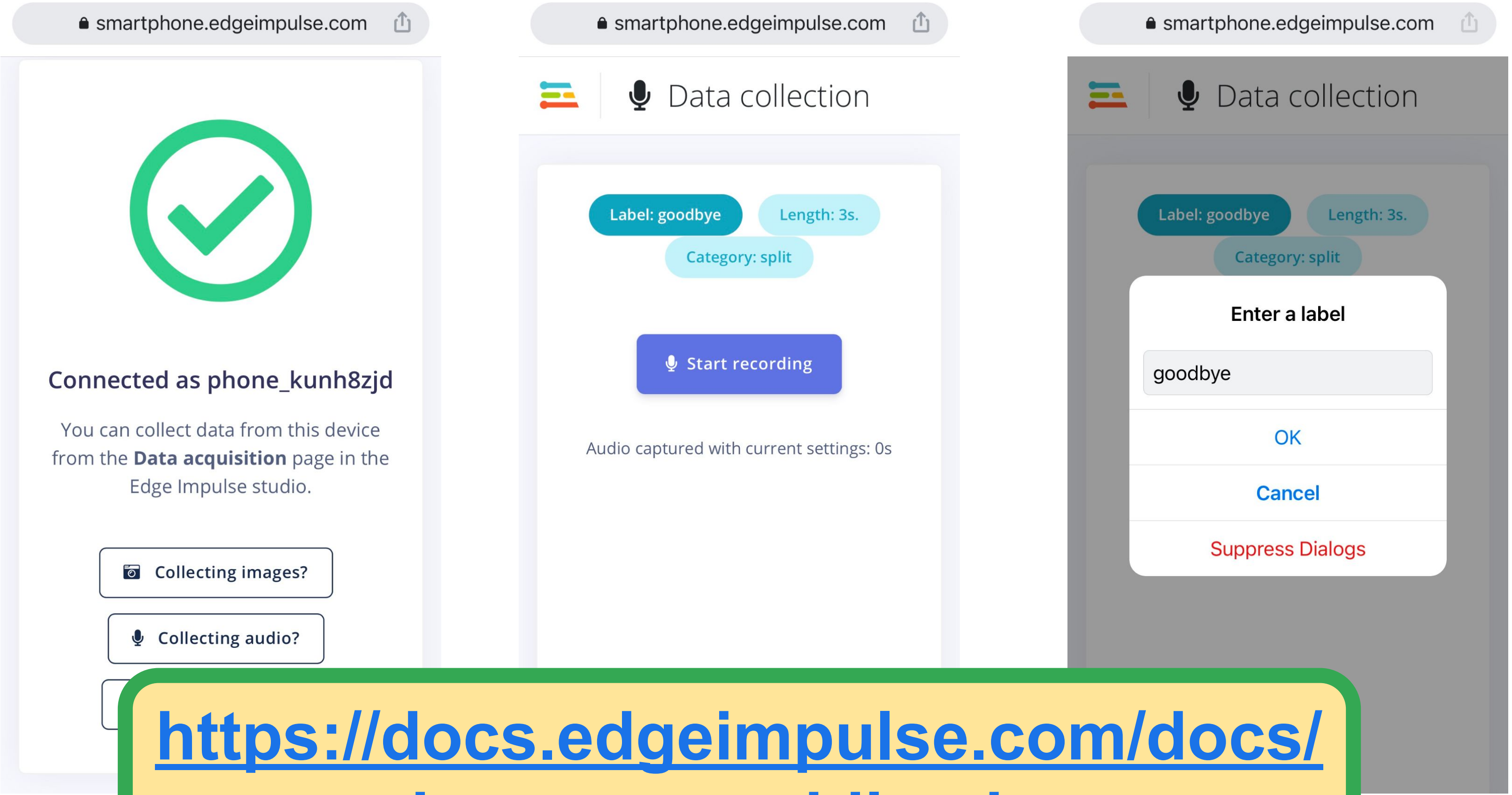

smartphone.edgeimpulse.com
Connected as phone_kunh8zjd
You can collect data from this device from the Data acquisition page in the Edge Impulse studio.
Collecting images?
Collecting audio?
Collecting motion?


smartphone.edgeimpulse.com
Data collection
Label: goodbye
Start recording
Audio captured with current settings: 0s

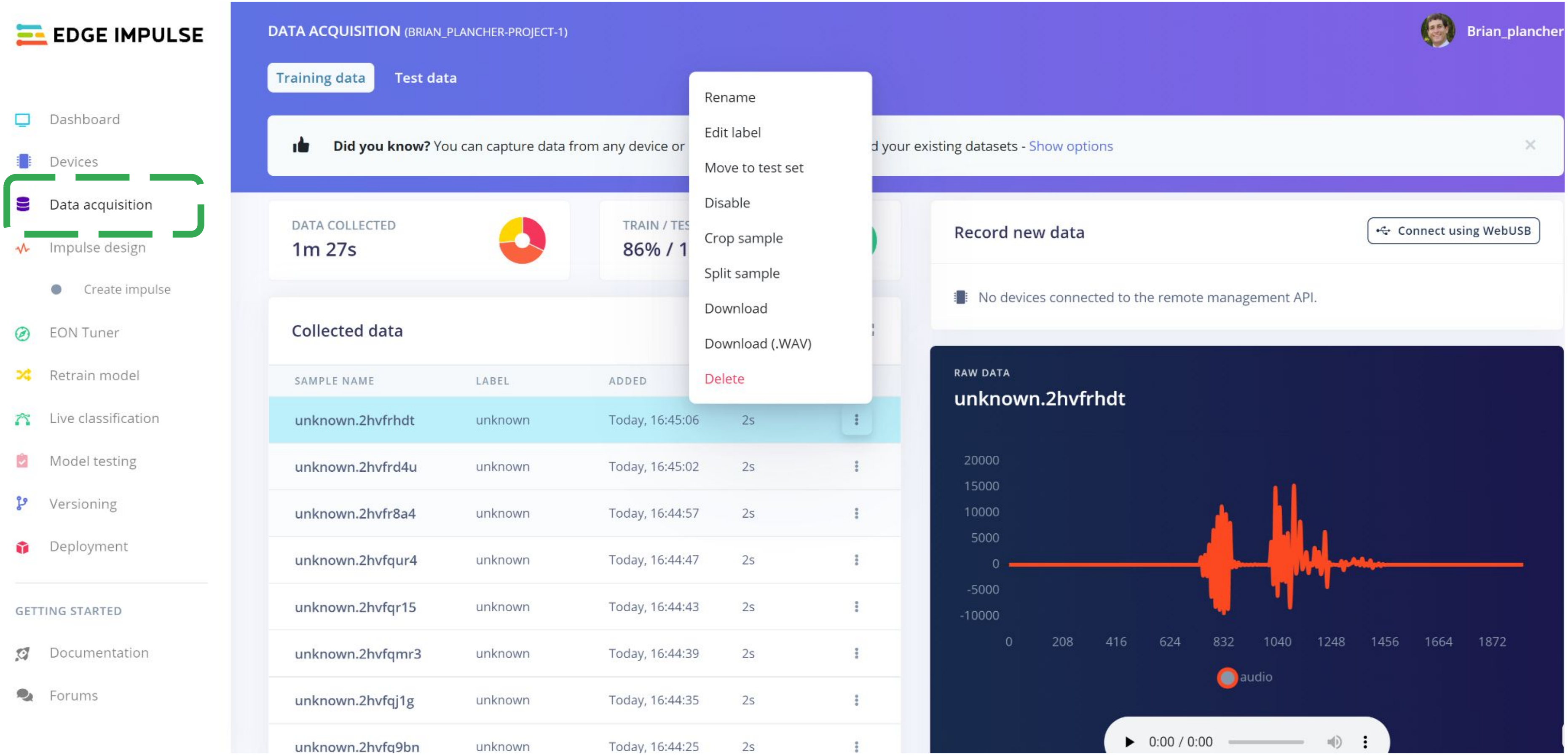

smartphone.edgeimpulse.com
Connected as phone_kunh8zjd
You can collect data from this device from the Data acquisition page in the Edge Impulse studio.
Collecting images?
Collecting audio?
Collecting motion?


smartphone.edgeimpulse.com
Data collection
Label: goodbye
Length: 3s.
Category: split
Start recording
Audio captured with current settings: 0s


smartphone.edgeimpulse.com
Data collection
Label: goodbye
Length: 3s.
Category: split
Enter a label
goodbye
OK
Cancel
Suppress Dialogs



https://docs.edgeimpulse.com/docs/using-your-mobile-phone


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 (BRIAN_PLANCHER-PROJECT-1)
Brian_plancher
Training data
Test data
Did you know? You can capture data from any device or
d your existing datasets - Show options
Rename
Edit label
Move to test set
Disable
Crop sample
Split sample
Download
Download (.WAV)
Delete
DATA COLLECTED
1m 27s
TRAIN / TES
86% / 1
Collected data
SAMPLE NAME
LABEL
ADDED
unknown.2hvfrhdt unknown Today, 16:45:06 2s
unknown.2hvfrd4u unknown Today, 16:45:02 2s
unknown.2hvfr8a4 unknown Today, 16:44:57 2s
unknown.2hvfqur4 unknown Today, 16:44:47 2s
unknown.2hvfqr15 unknown Today, 16:44:43 2s
unknown.2hvfqmr3 unknown Today, 16:44:39 2s
unknown.2hvfqj1g unknown Today, 16:44:35 2s
unknown.2hvfq9bn unknown Today, 16:44:25 2s
Record new data
Connect using WebUSB
No devices connected to the remote management API.
RAW DATA
unknown.2hvfrhdt
20000
15000
10000
5000
0
-5000
-10000
0
208
416
624
832
1040
1248
1456
1664
1872
audio
0:00 / 0:00

https://docs.edgeimpulse.com/docs/using-your-mobile-phone

Activity: Create a Keyword Spotting Dataset

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

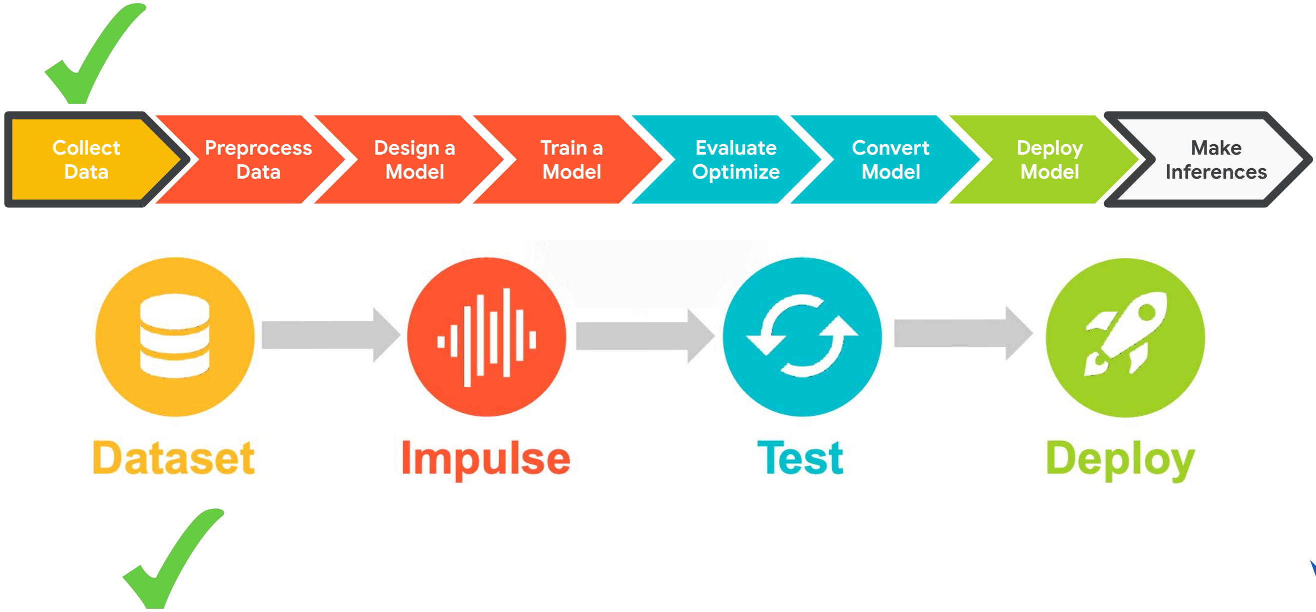
- **Keyword #1 “yes”** (label: yes) (length: 2 seconds)
- **Keyword #2 “no”** (label: no) (length: 2 seconds)
- **“Unknown” words** that are not the keyword **and background noise** (label: unknown) (length: 2 seconds)

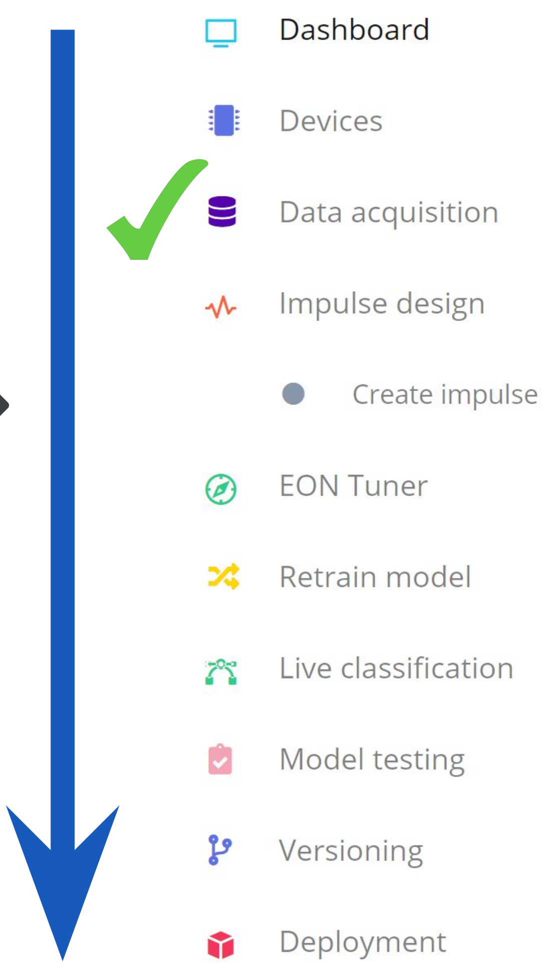
Also take a quick break! We’ll resume in 10 minutes!

Today’s Agenda

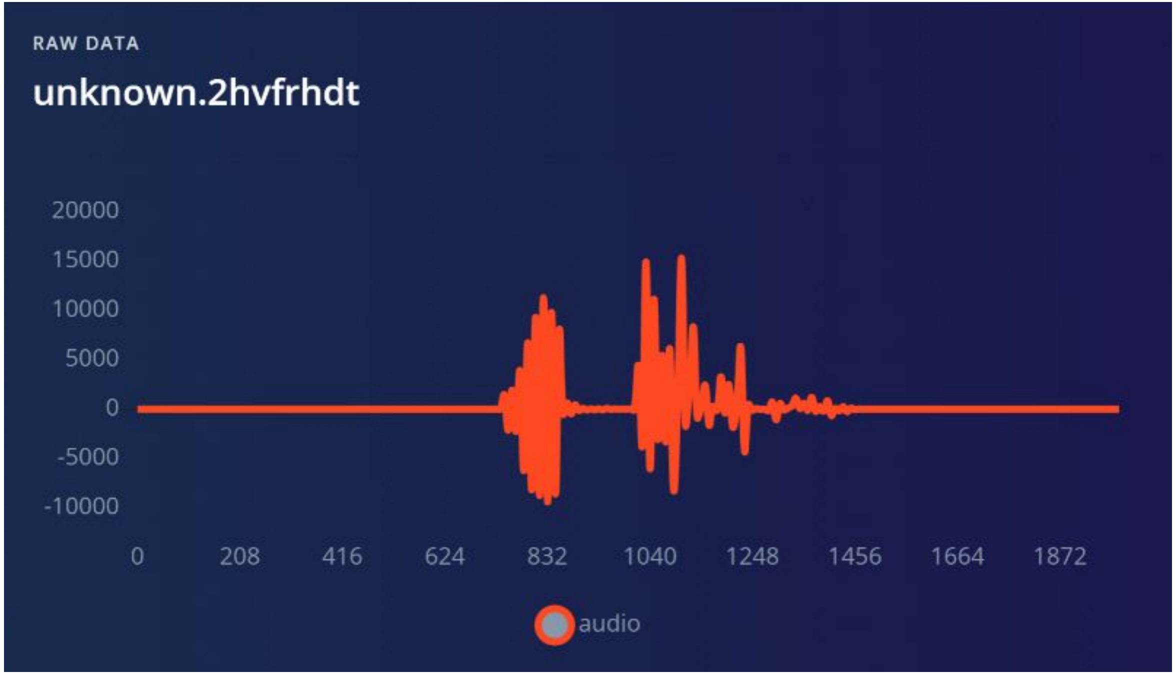
- Deep ML Background
- Hands-on Computer Vision: Thing Translator
- The Tiny Machine Learning Workflow
- Keyword Spotting (KWS) Data Collection
- KWS Preprocessing and Training
    - **Preprocessing (for KWS)**
    - Hands-on Preprocessing and Training with Edge Impulse
- Deployment Challenges and Opportunities for Embedded ML
- Summary

Edge Impulse Project Dashboard



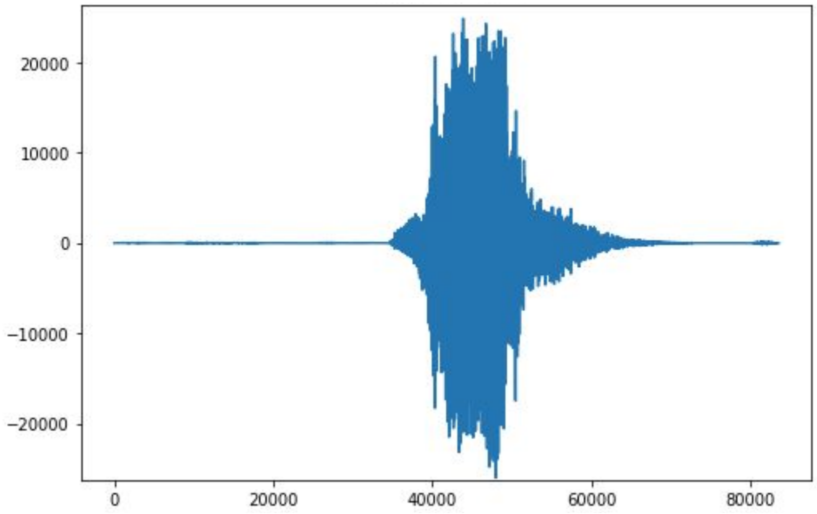


Why might we want to **preprocess** data and not send the raw data to the neural network?

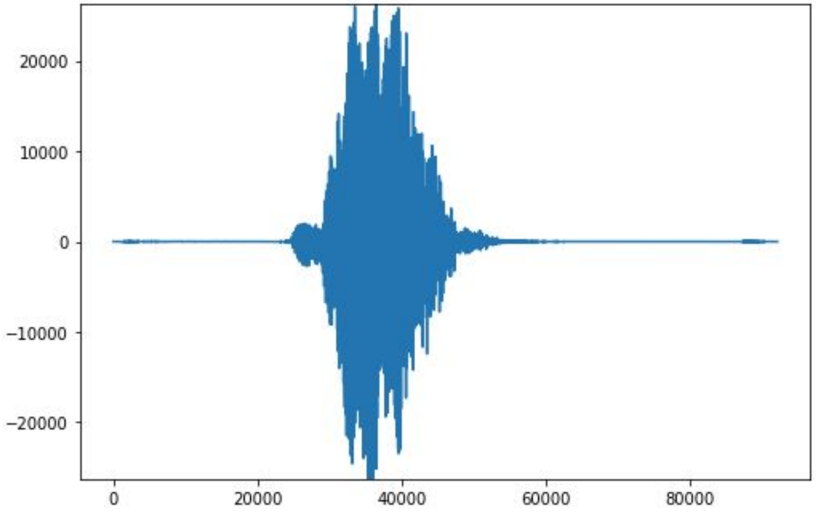


Can you tell these two signals apart?

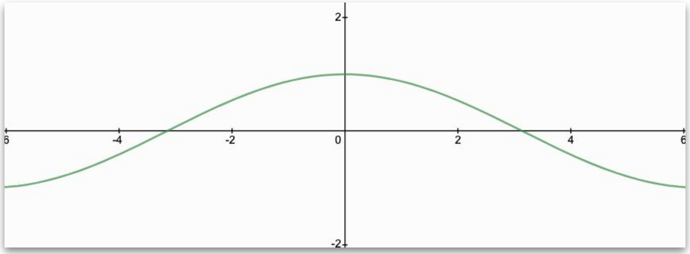
“Yes” *(spoken loudly)*



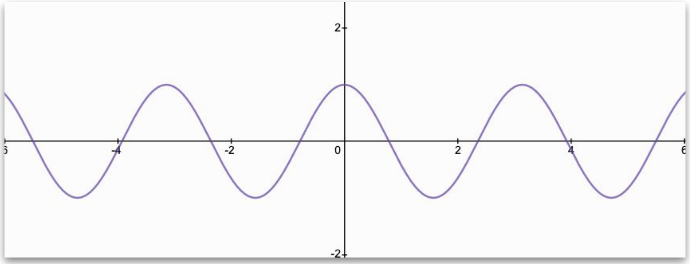
“No” *(spoken loudly)*



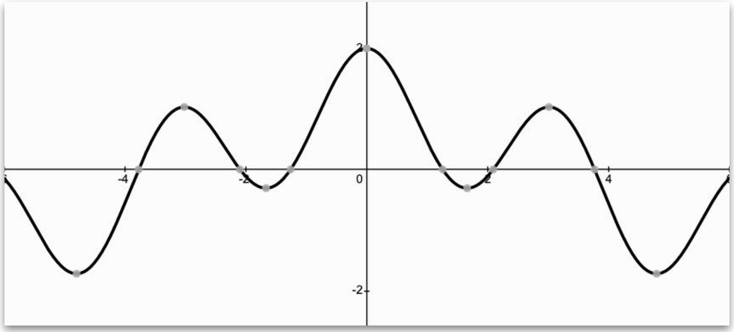
Signal **Components**?



+



=



Signal **Components**?

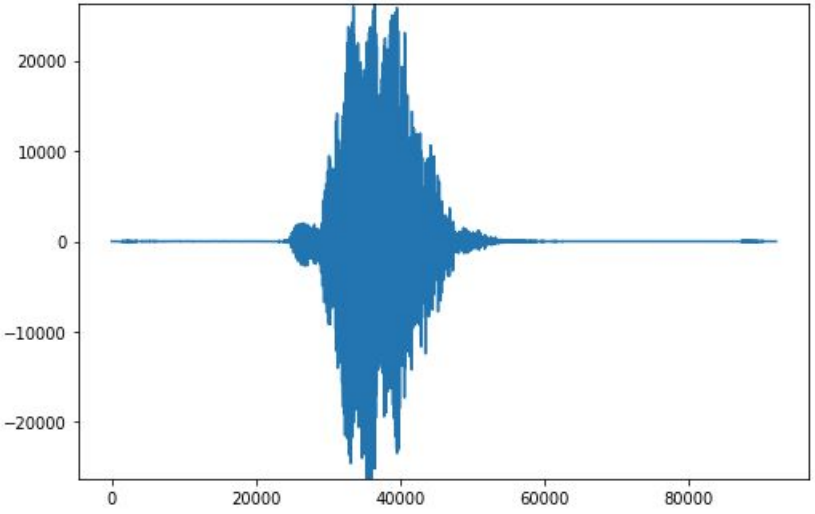
?

+

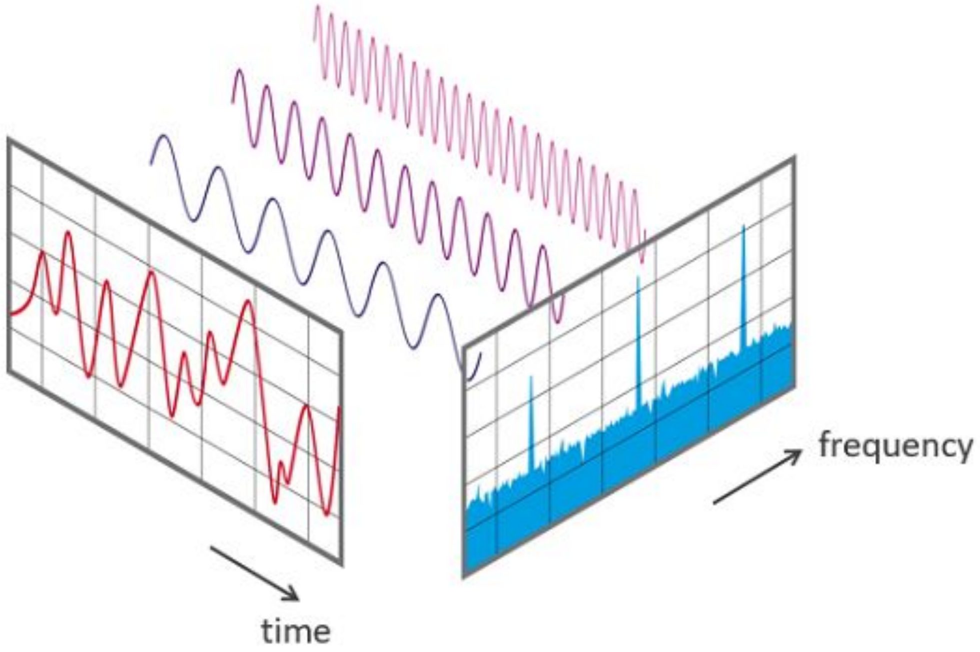
?

=

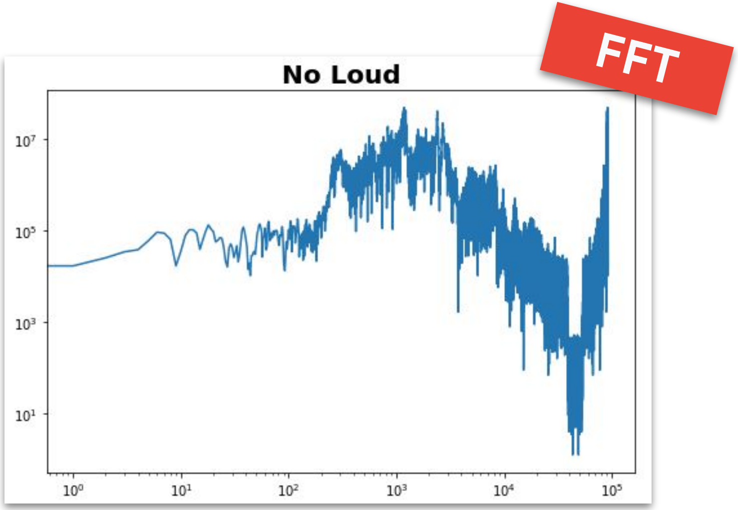
“No” *(spoken loudly)*



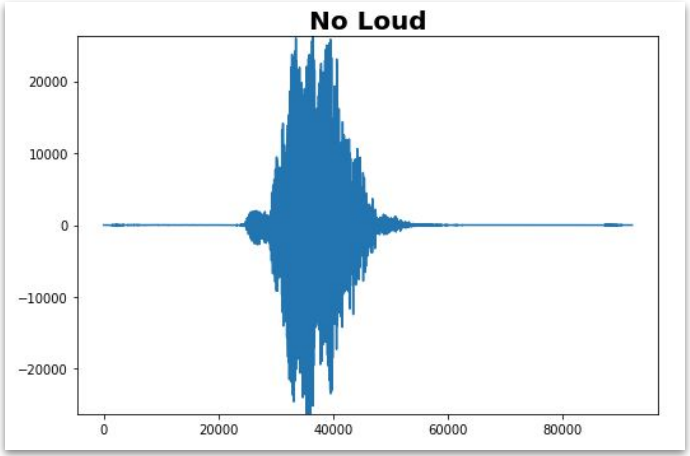
Fast Fourier Transform:
extract the frequencies from a signal



Fast Fourier Transform

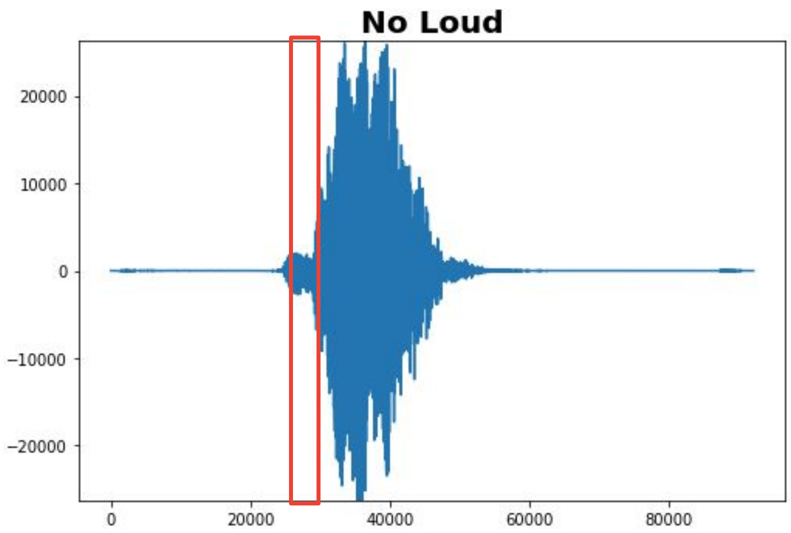


Frequency

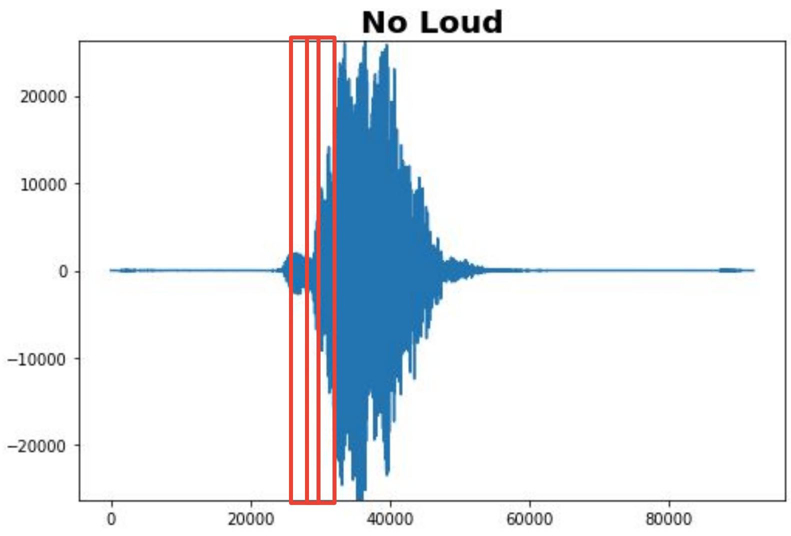


Time

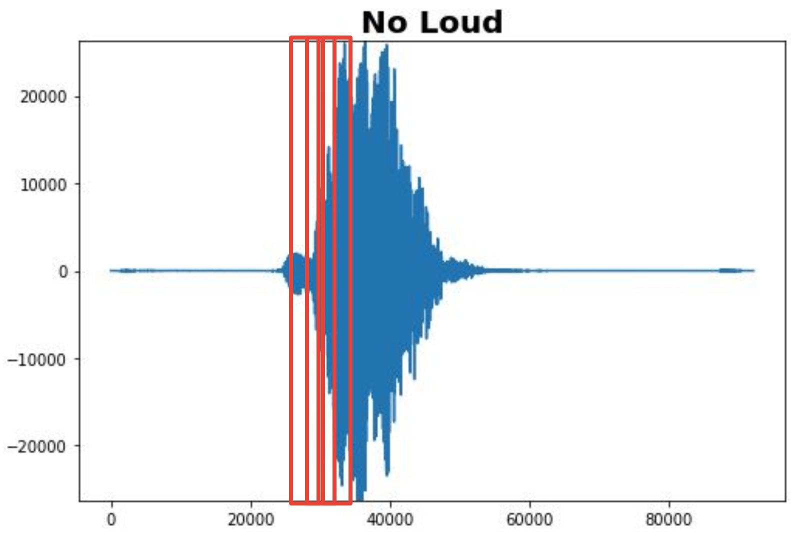
Building a **Spectrogram** using FFTs



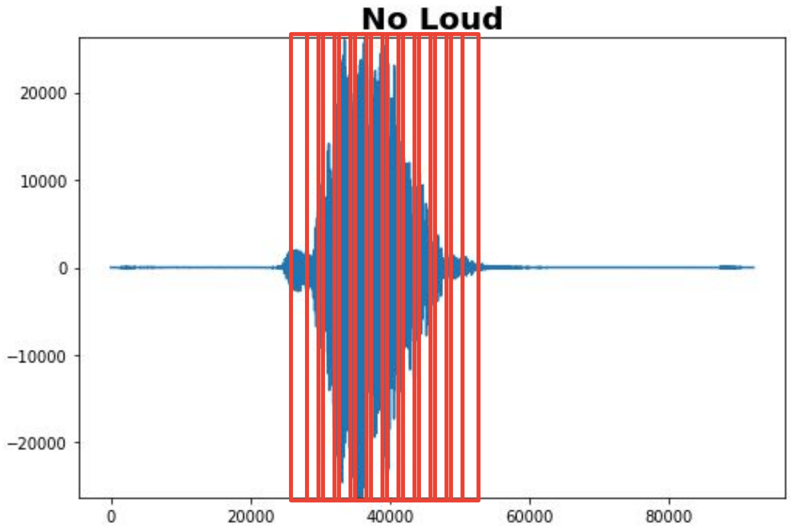
Building a **Spectrogram** using FFTs

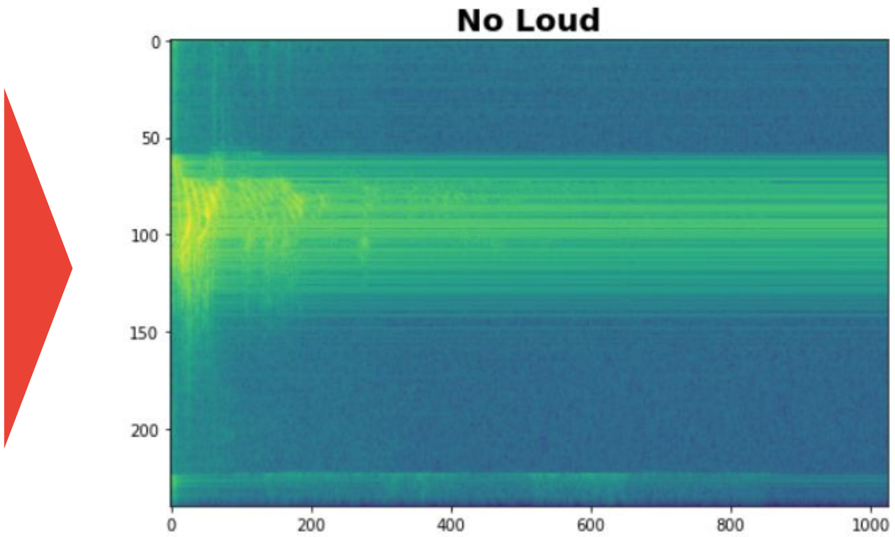


Building a **Spectrogram** using FFTs



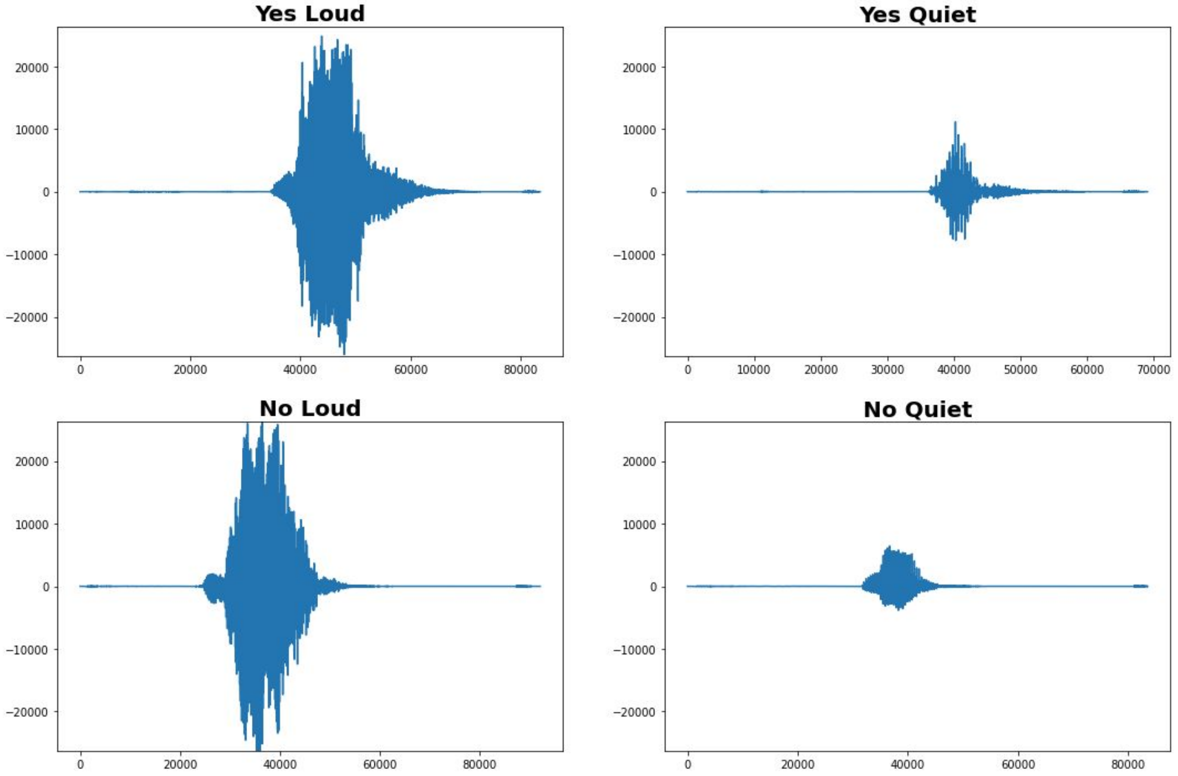
Building a **Spectrogram** using FFTs



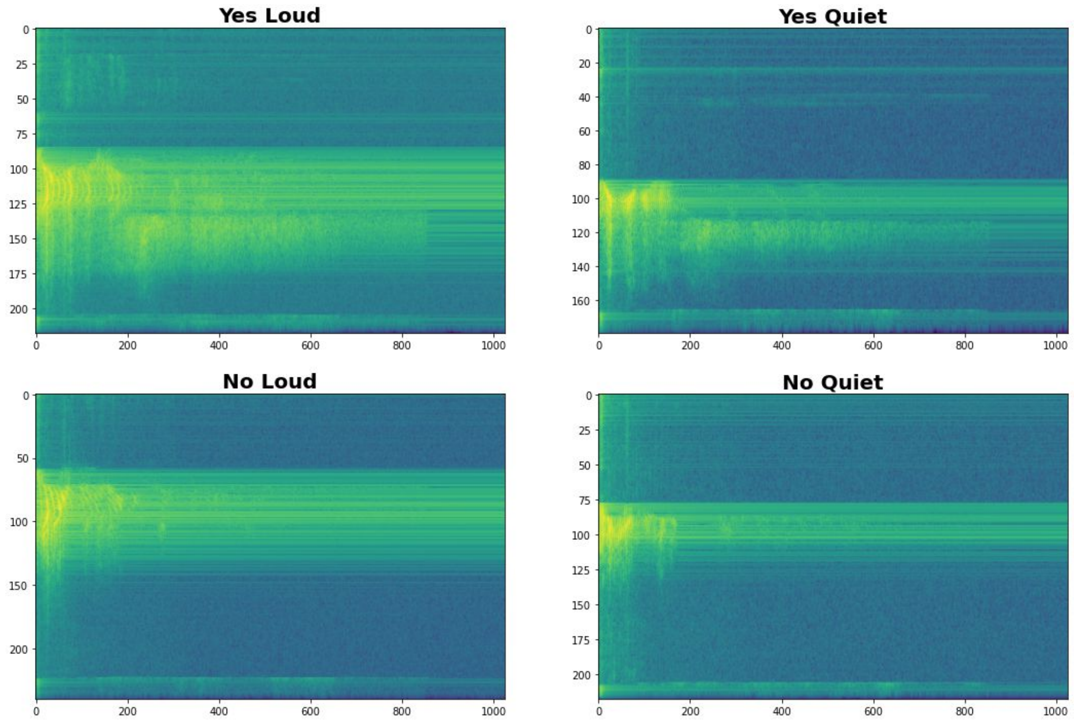


Essentially if you **stack up all the FFTs in a row** then you get the **Spectrogram** (time vs. frequency with color indicating intensity)

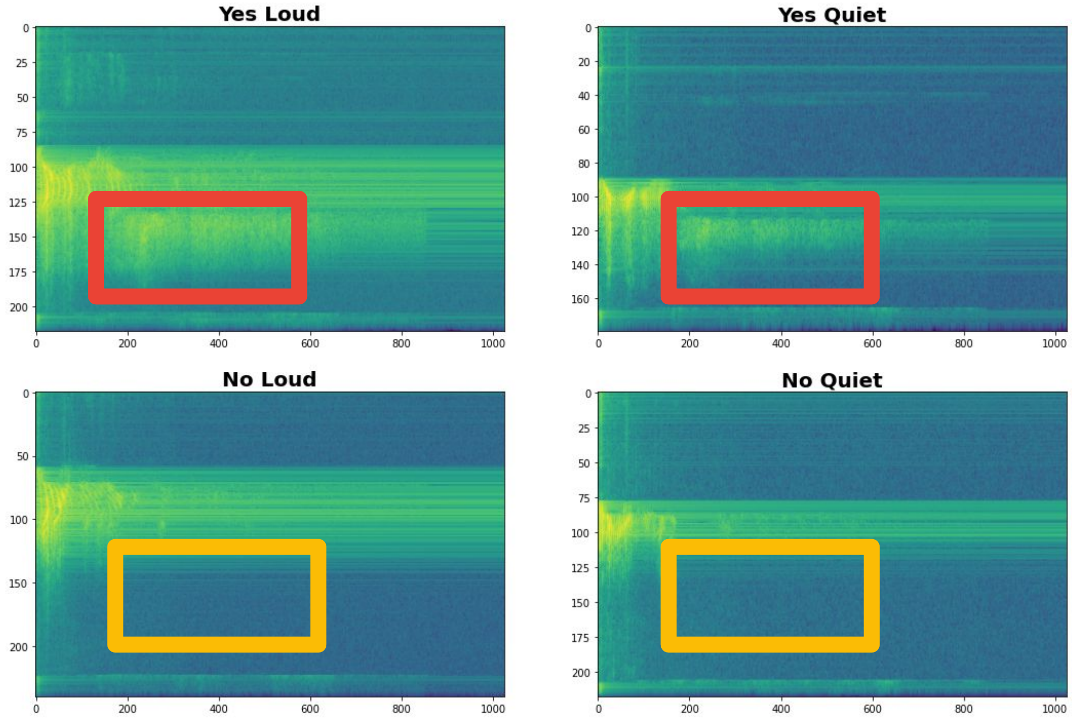
Spectrograms help differentiate the data



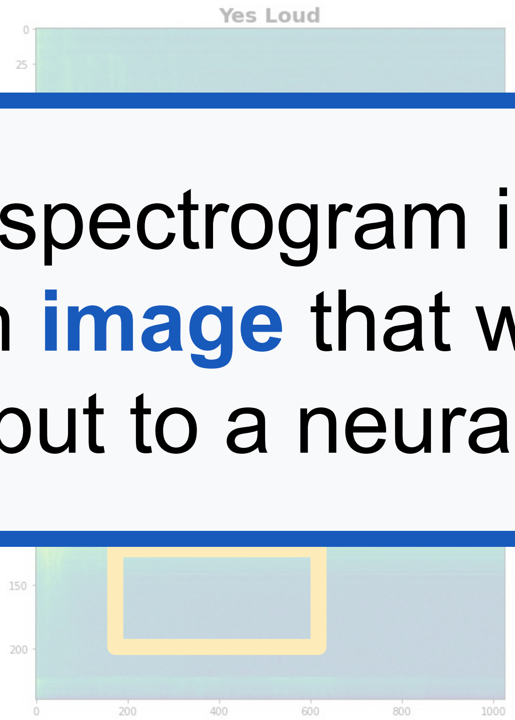
Spectrograms help differentiate the data

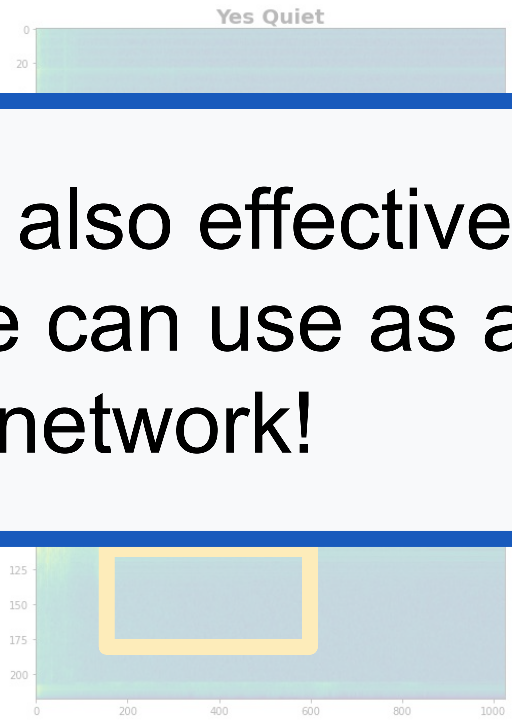


Spectrograms help differentiate the data



Data Preprocessing: **Spectrograms**

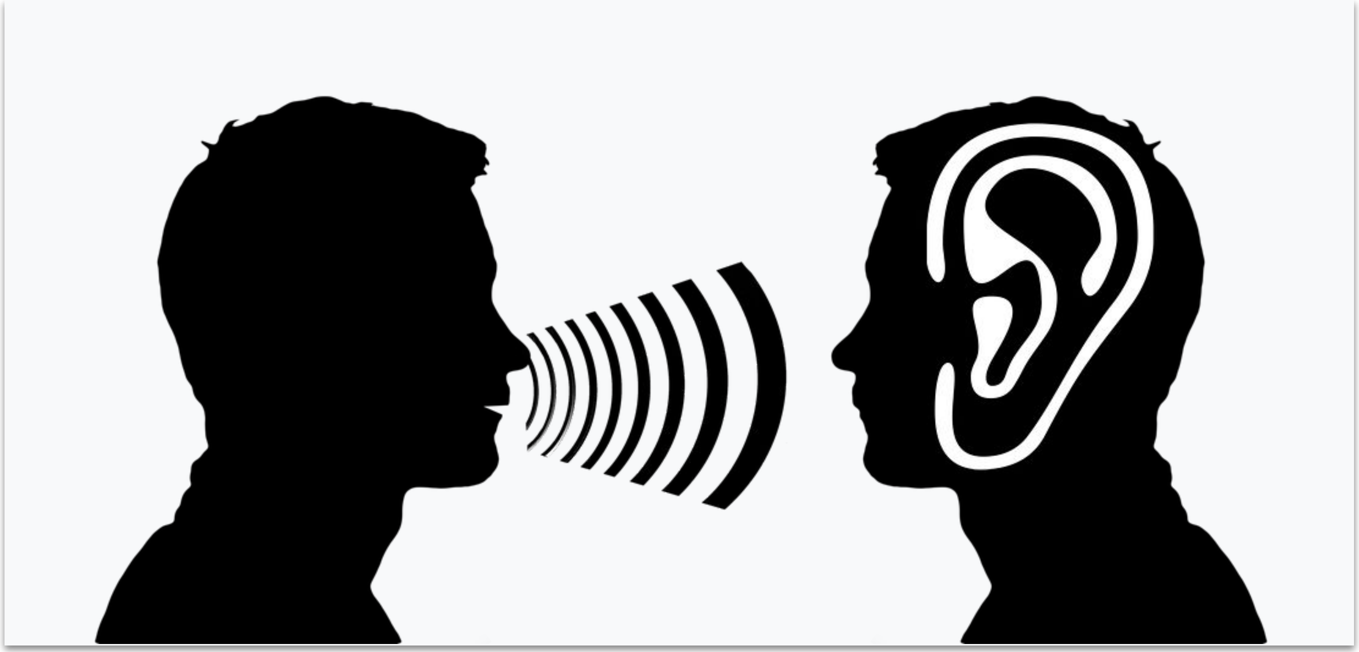




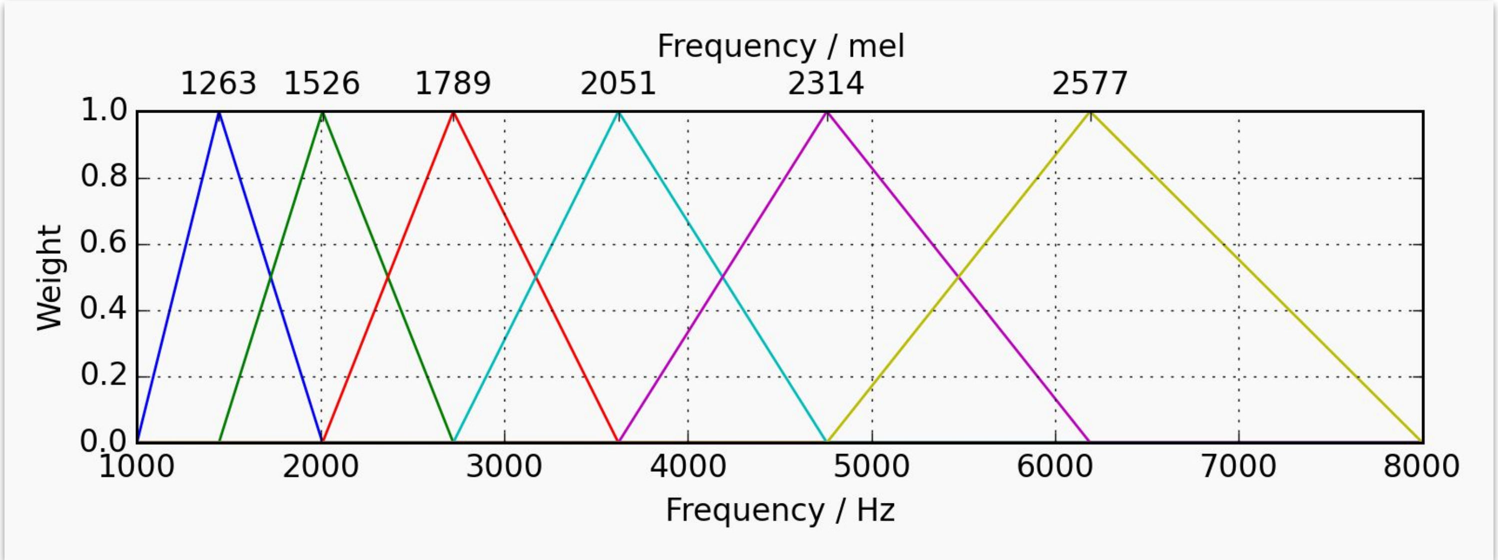
A spectrogram is also effectively an **image** that we can use as an input to a neural network!

Can we do **better** than a spectrogram?

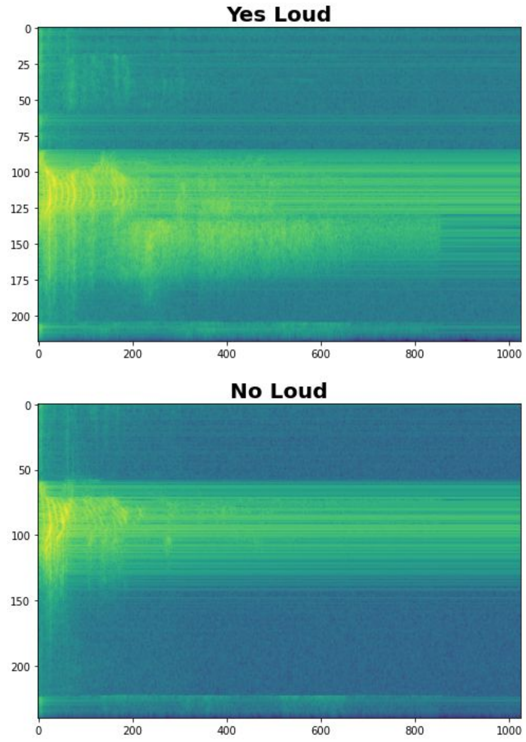
Can we take **domain knowledge** into account?

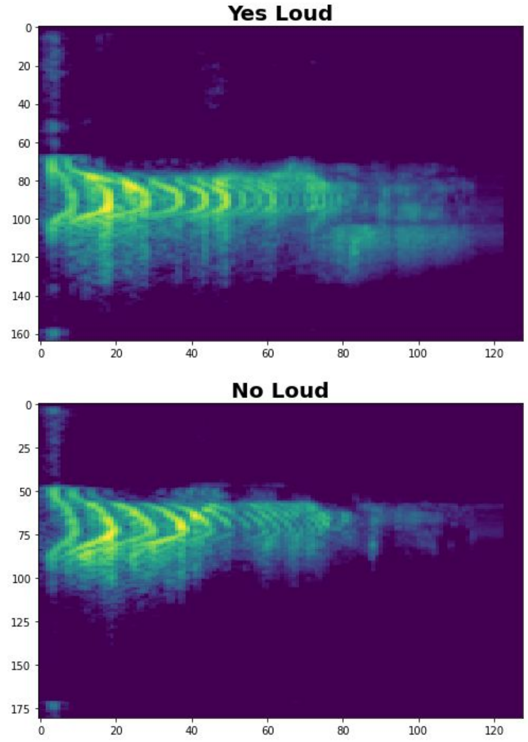


Mel Filterbanks

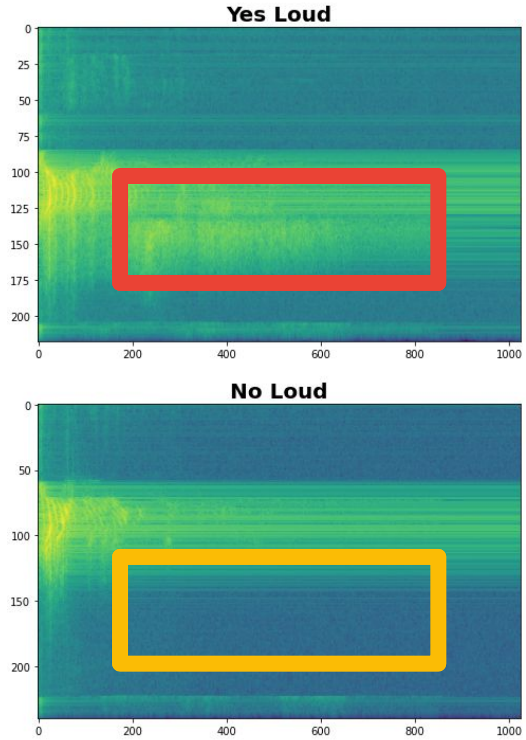


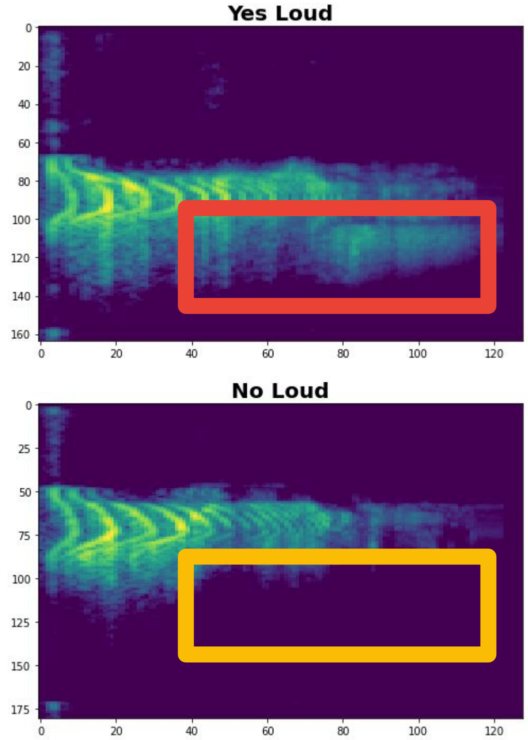
Spectrograms v. MFCCs





Spectrograms v. MFCCs



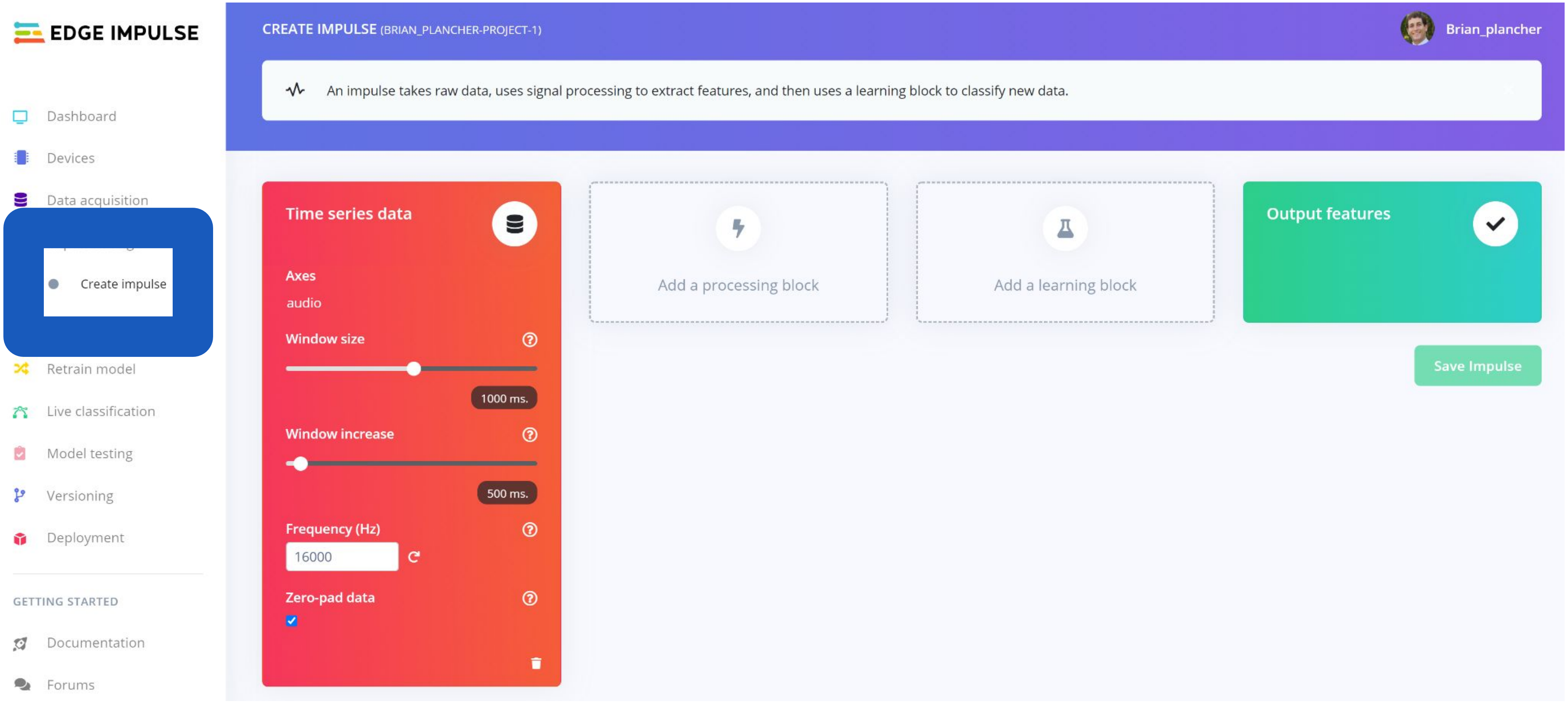


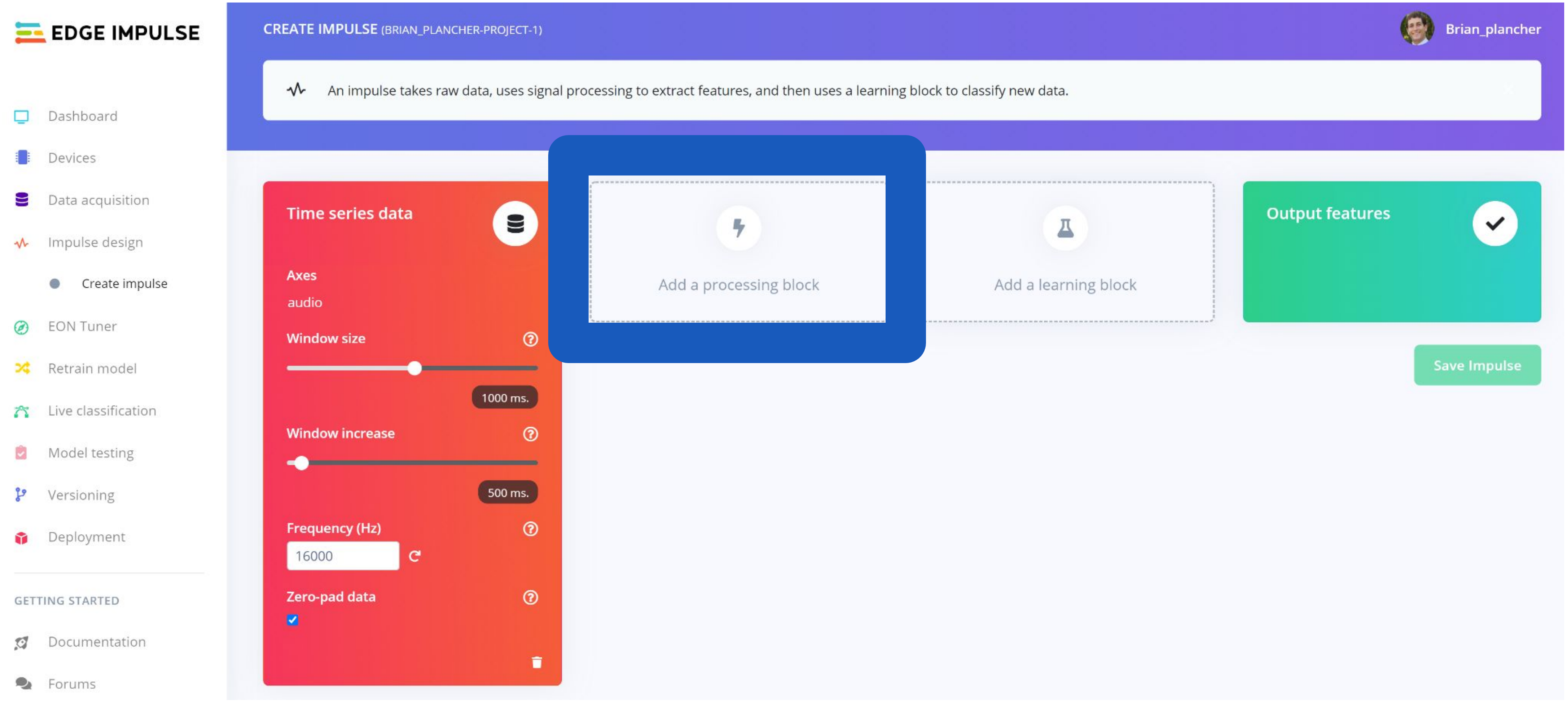
Additional **Feature Engineering**

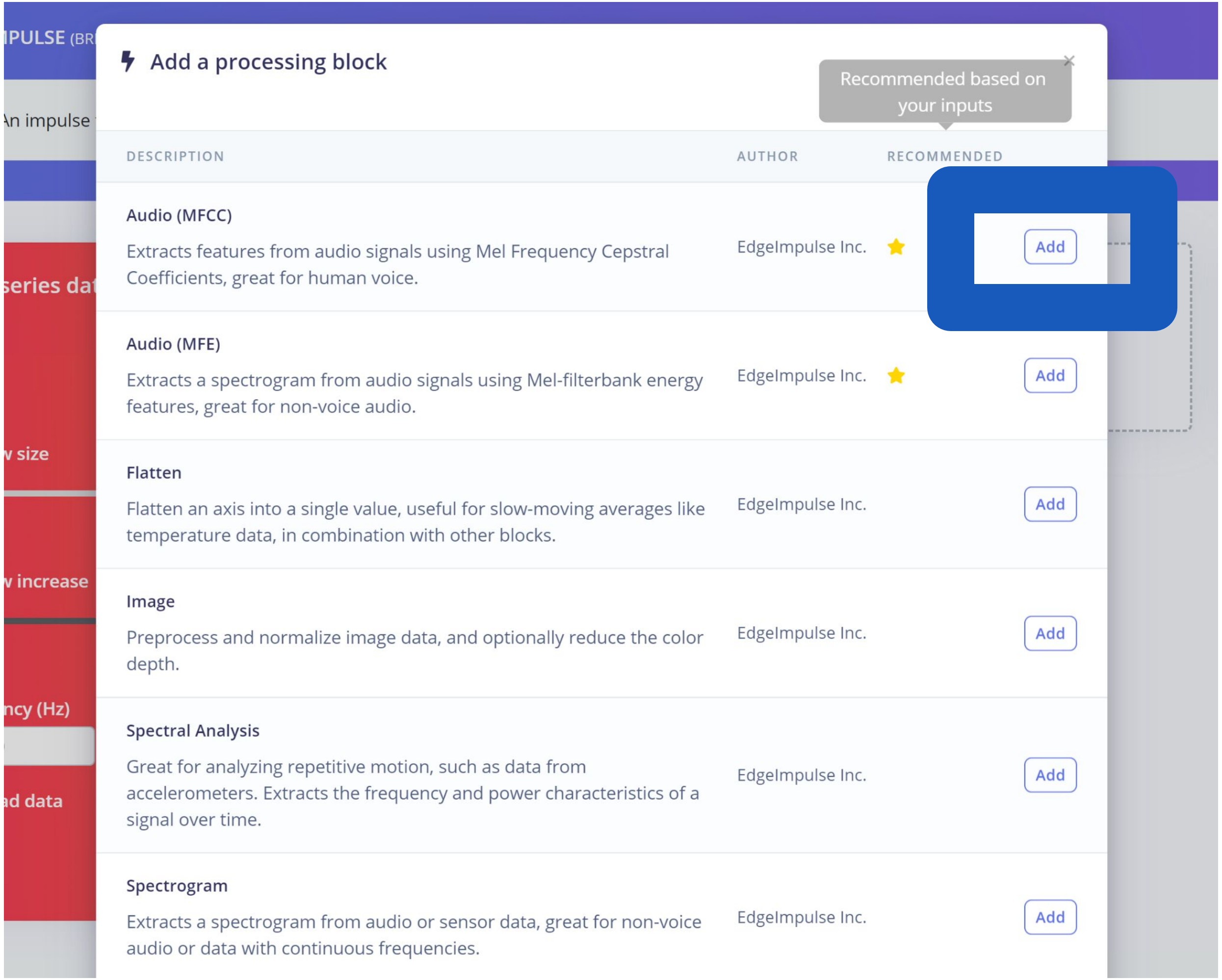
WARNING: Whatever preprocessing you do on the computer in python for training you need to do in C++ on the microcontroller!

Today’s Agenda

- Deep ML Background
- Hands-on Computer Vision: Thing Translator
- The Tiny Machine Learning Workflow
- Keyword Spotting (KWS) Data Collection
- KWS Preprocessing and Training
    - Preprocessing (for KWS)
    - **Hands-on Preprocessing and Training with Edge Impulse**
- Deployment Challenges and Opportunities for Embedded ML
- Summary


EDGE IMPULSE
Dashboard
Devices
Data acquisition
Create impulse
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
CREATE IMPULSE (BRIAN_PLANCHER-PROJECT-1)
Brian_plancher
An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.
Time series data
Axes
audio
Window size
1000 ms.
Window increase
500 ms.
Frequency (Hz)
16000
Zero-pad data
Add a processing block
Add a learning block
Output features
Save Impulse

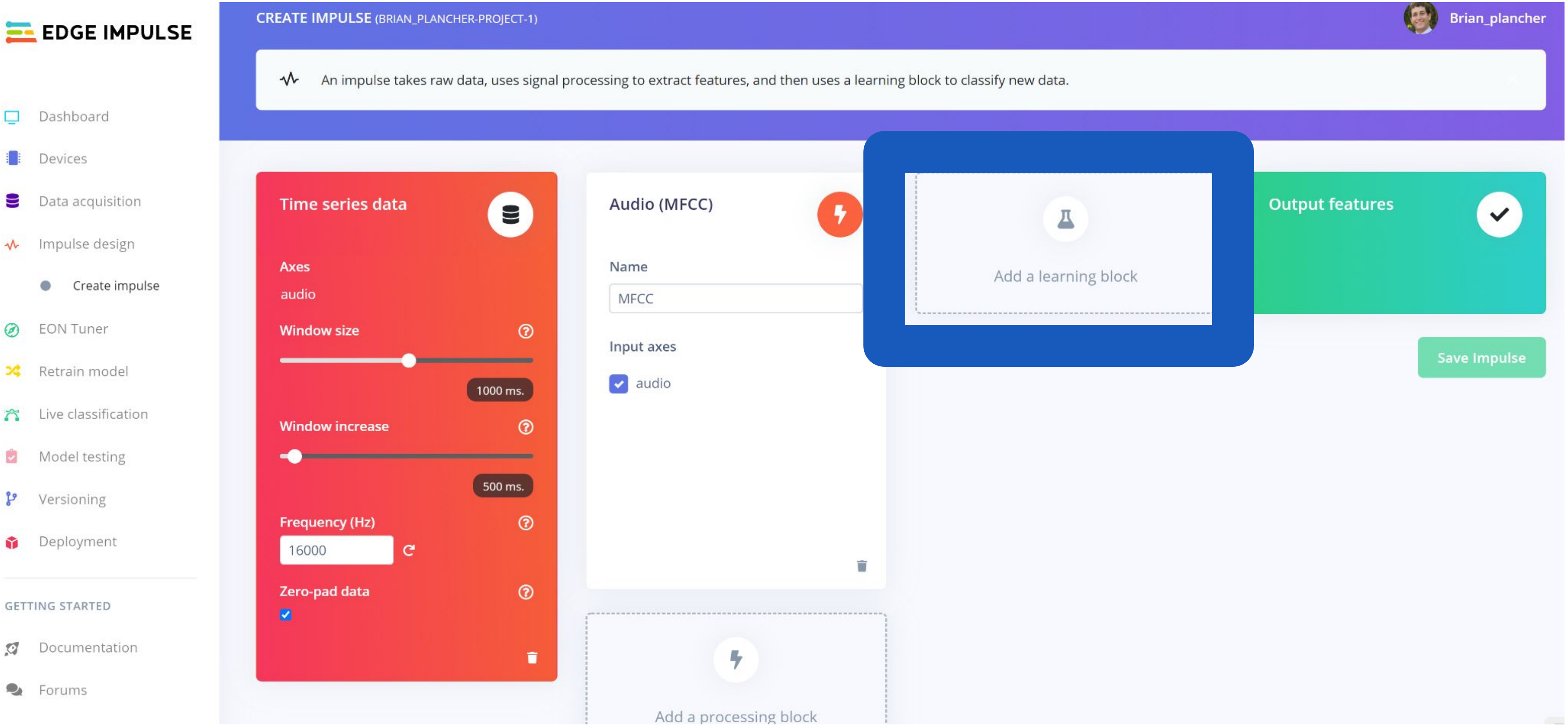

EDGE IMPULSE
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
CREATE IMPULSE (BRIAN_PLANCHER-PROJECT-1)
Brian_plancher
An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.
Time series data
Axes
audio
Window size
1000 ms.
Window increase
500 ms.
Frequency (Hz)
16000
Zero-pad data
Add a processing block
Add a learning block
Output features
Save Impulse

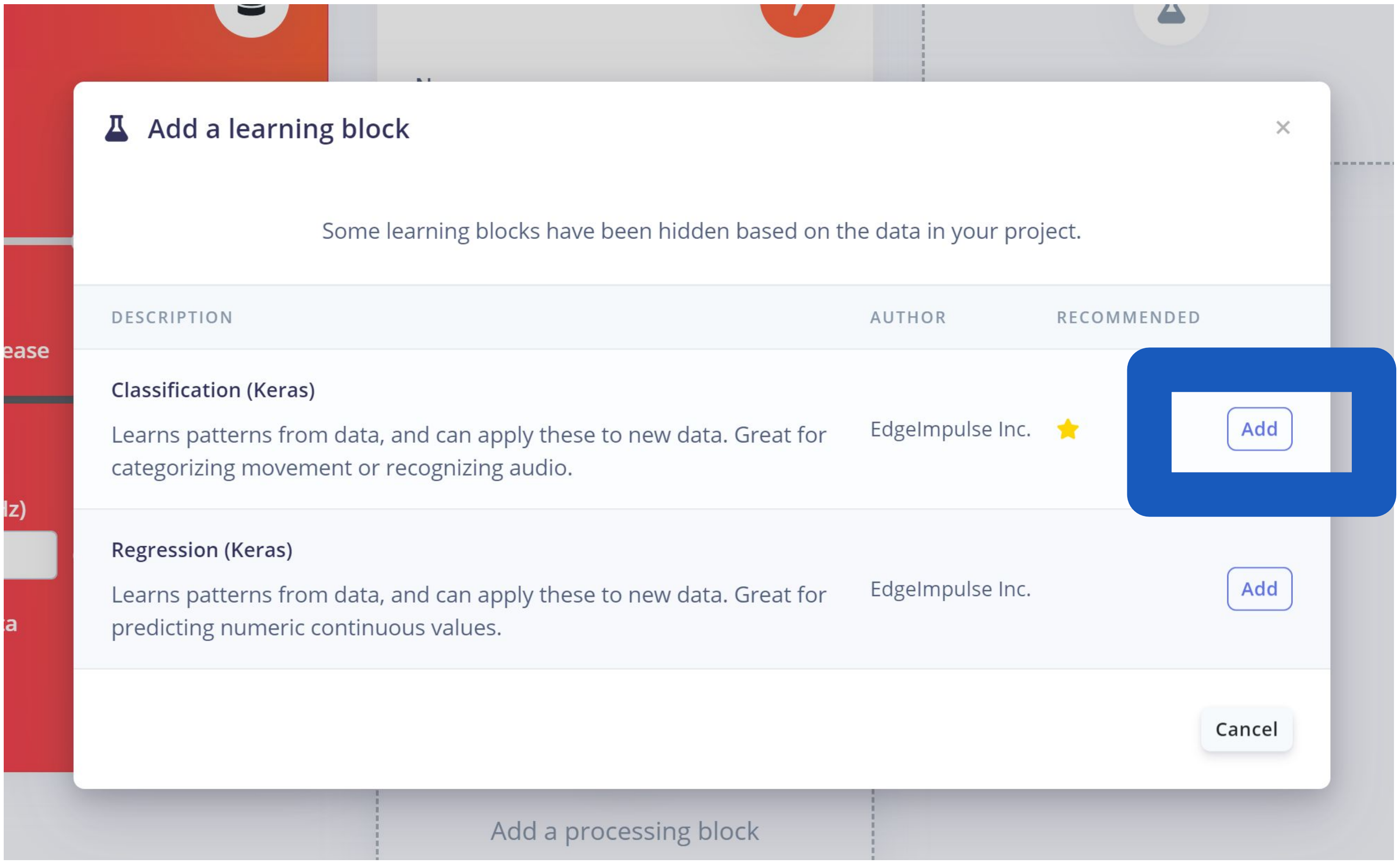


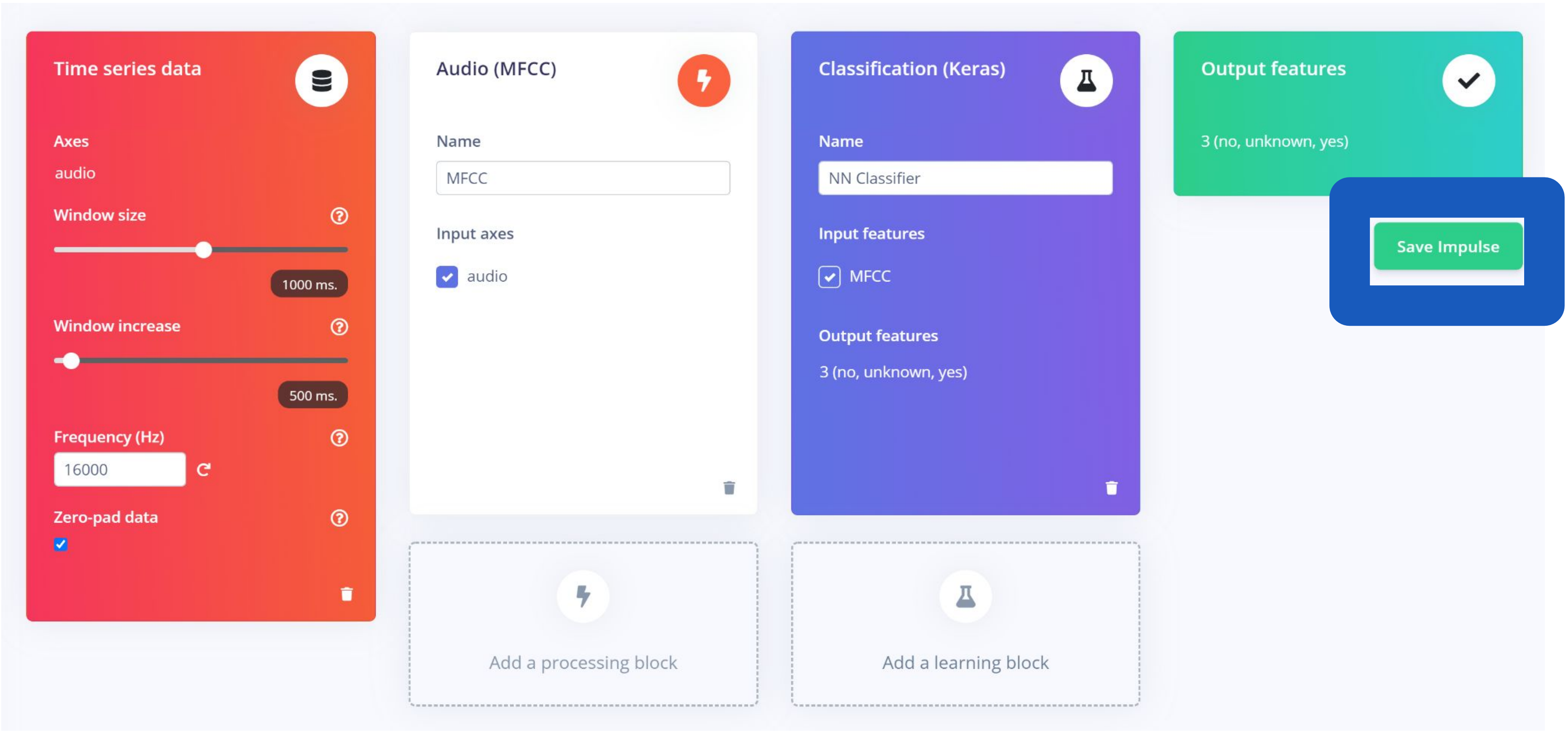
We'll keep things simple today and just add an MFCC but/and in future projects you can:

- **create your own blocks**
- **use multiple blocks**

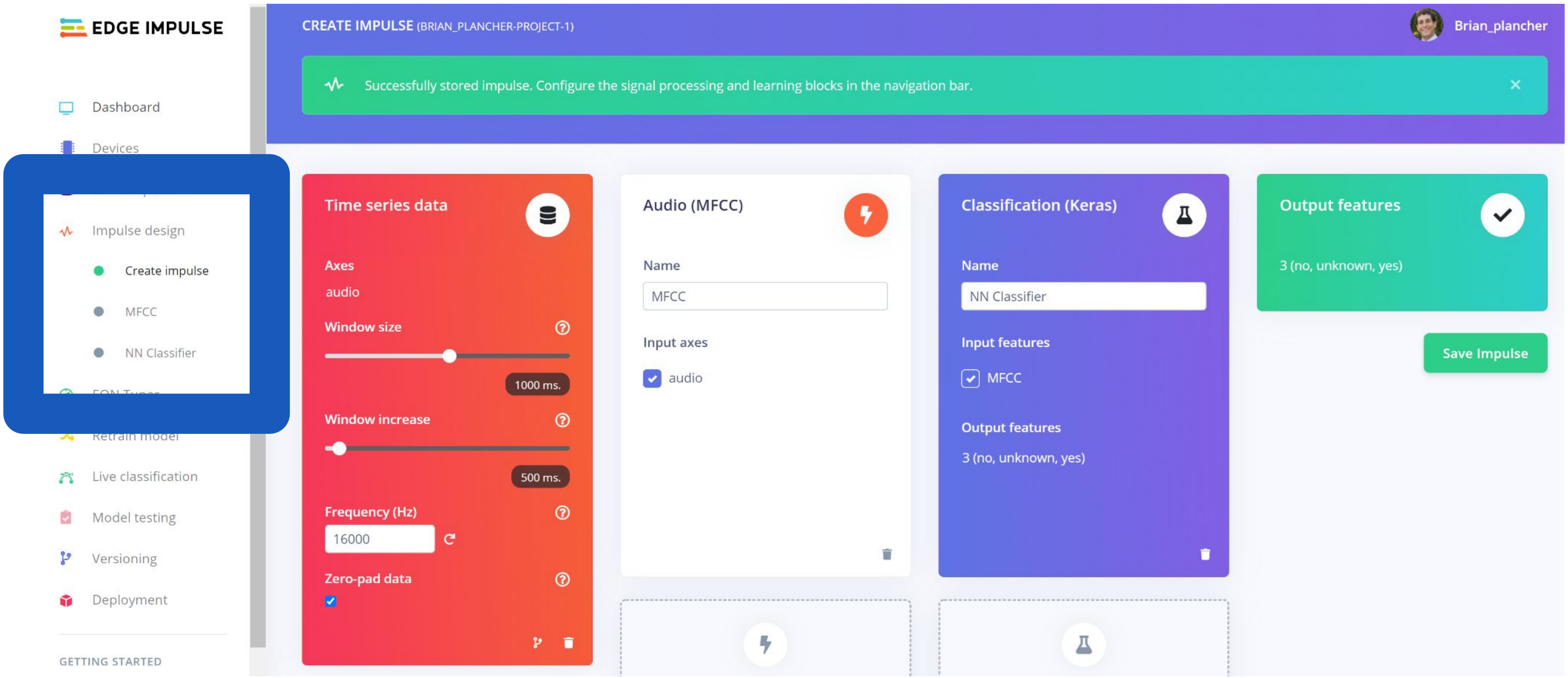
https://docs.edgeimpulse.com/docs/custom-blocks

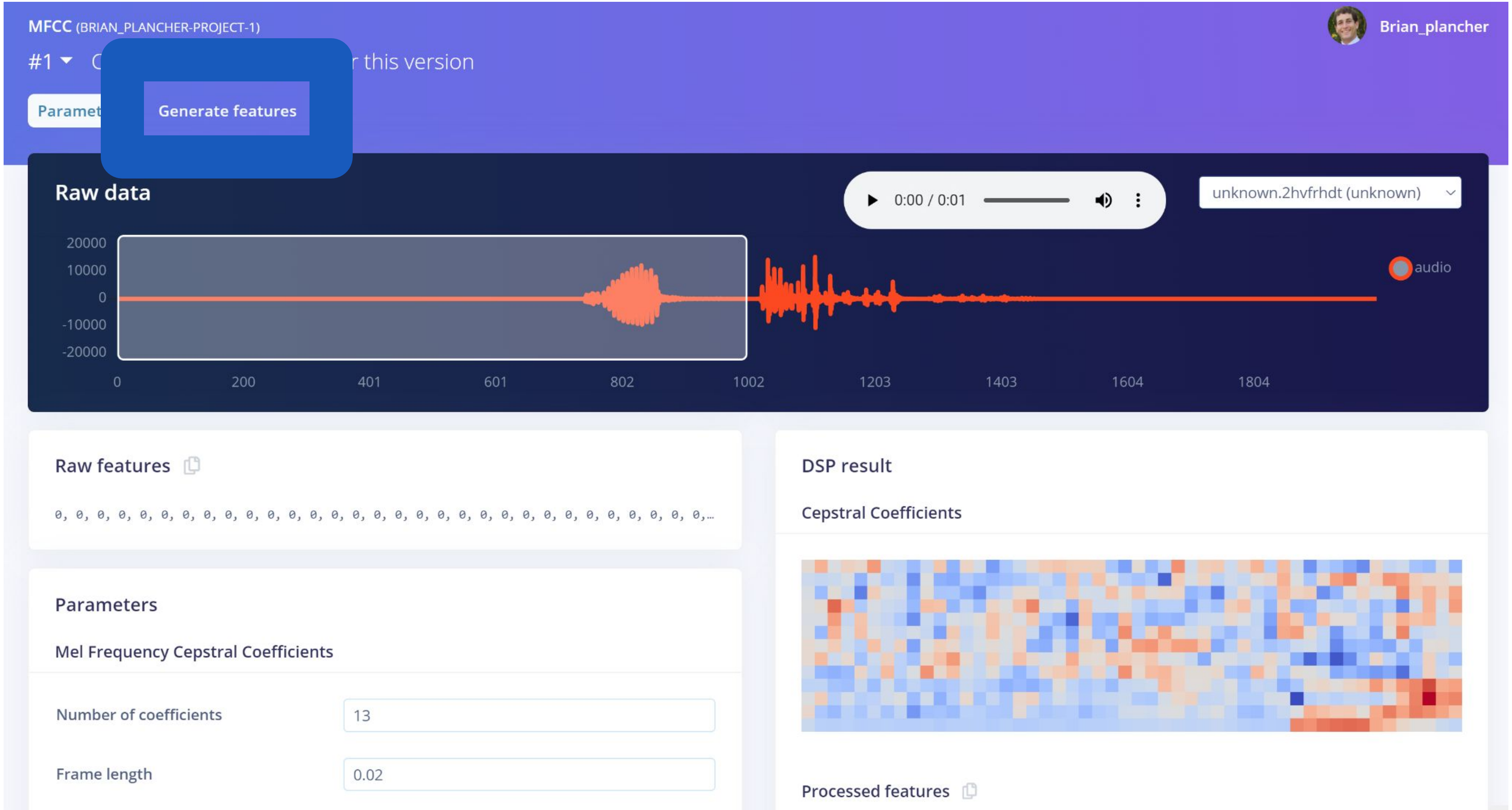

EDGE IMPULSE
CREATE IMPULSE (BRIAN_PLANCHER-PROJECT-1)
Brian_plancher
An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
Time series data
Axes
audio
Window size
1000 ms.
Window increase
500 ms.
Frequency (Hz)
16000
Zero-pad data
Audio (MFCC)
Name
MFCC
Input axes
audio
Add a processing block
Add a learning block
Output features
Save Impulse

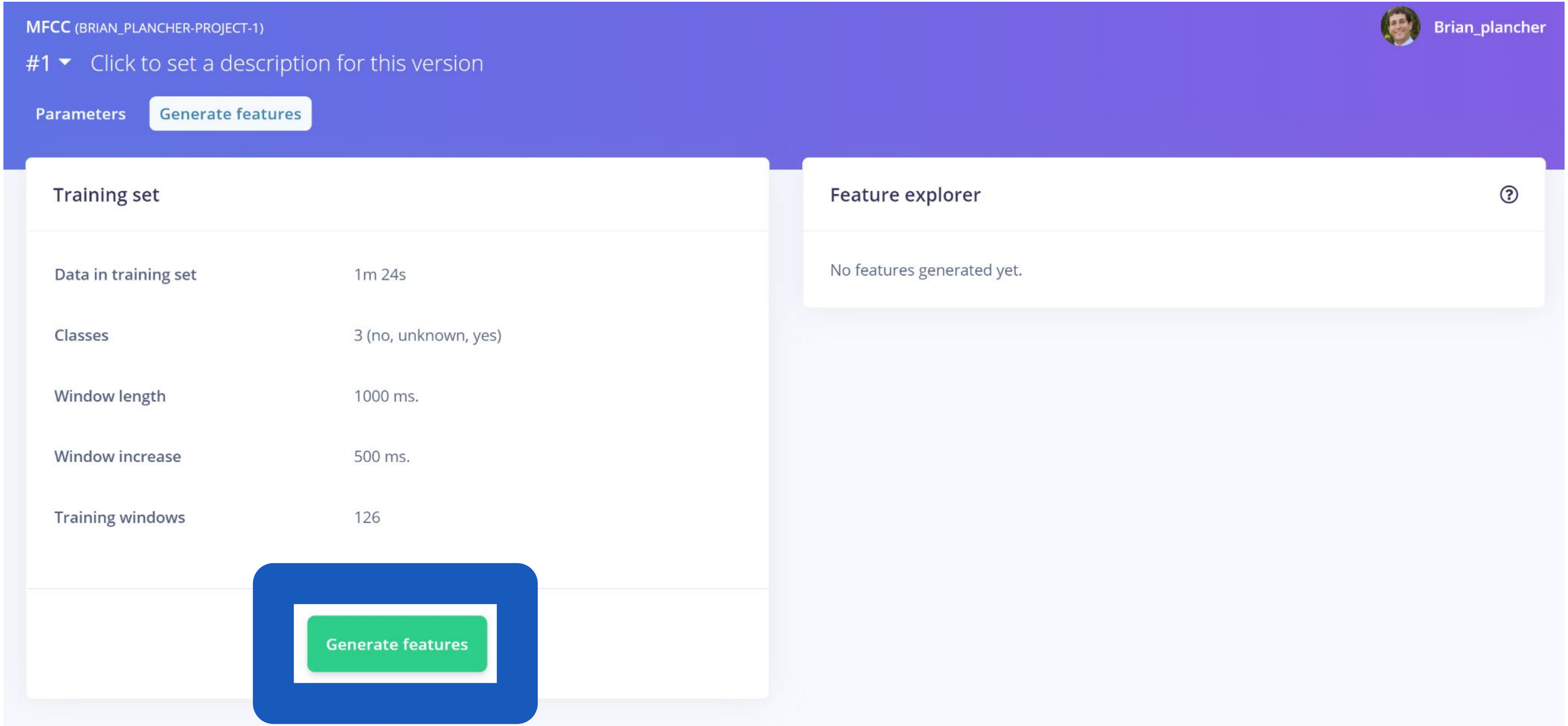

Add a learning block
Some learning blocks have been hidden based on the data in your project.
DESCRIPTION
AUTHOR
RECOMMENDED
Classification (Keras)
Learns patterns from data, and can apply these to new data. Great for categorizing movement or recognizing audio.
EdgeImpulse Inc.
Add
Regression (Keras)
Learns patterns from data, and can apply these to new data. Great for predicting numeric continuous values.
EdgeImpulse Inc.
Add
Cancel
Add a processing block

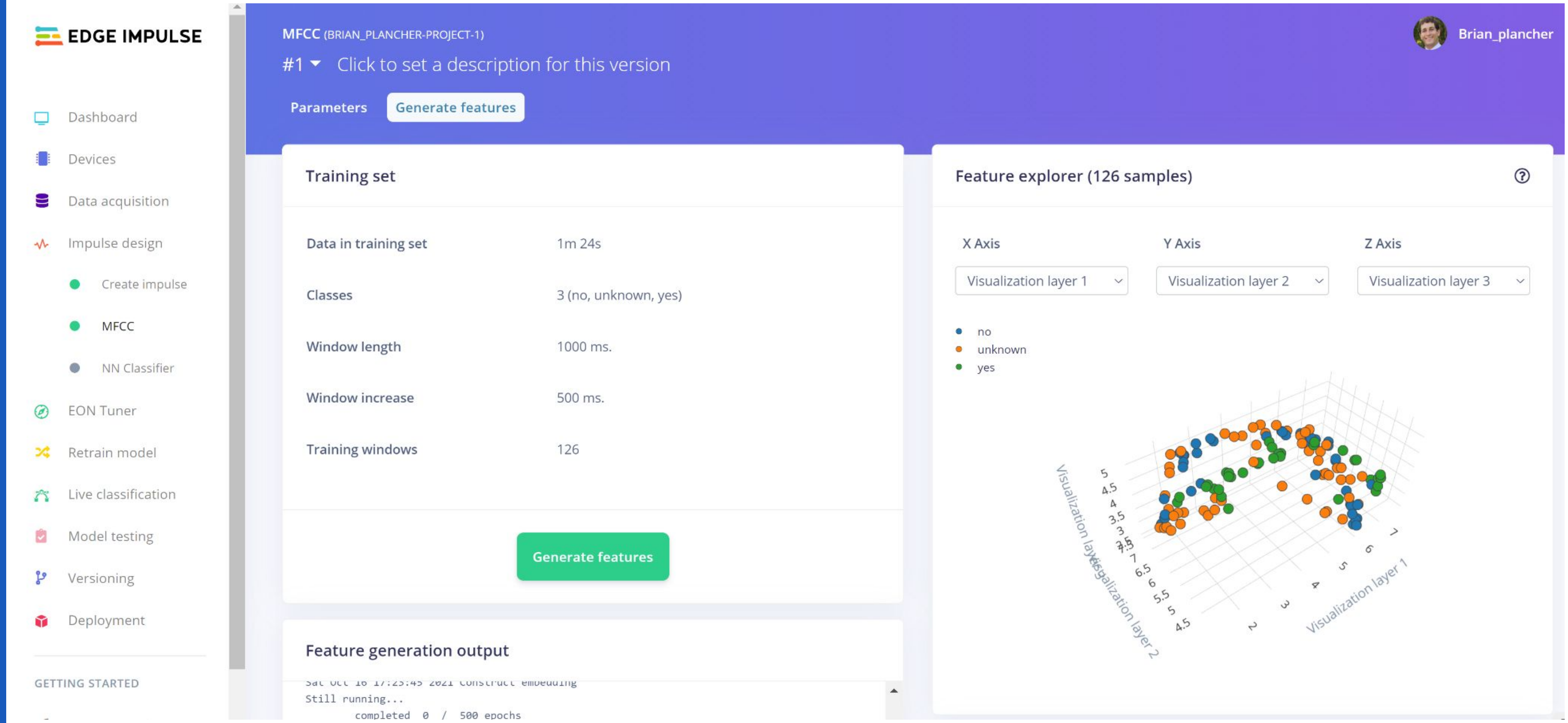


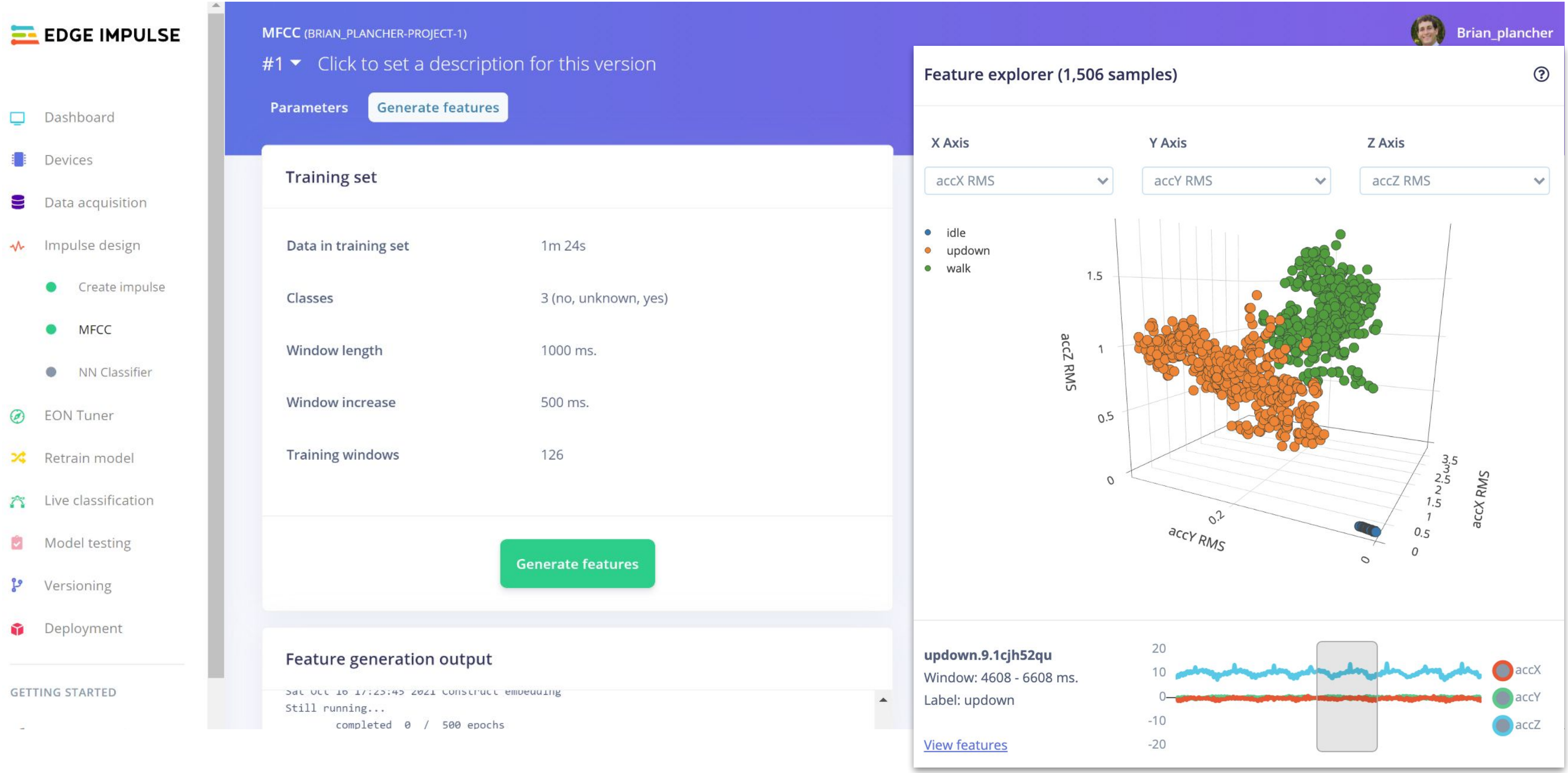
Time series data
Axes
audio
Window size
1000 ms.
Window increase
500 ms.
Frequency (Hz)
16000
Zero-pad data
Audio (MFCC)
Name
MFCC
Input axes
audio
Classification (Keras)
Name
NN Classifier
Input features
MFCC
Output features
3 (no, unknown, yes)
Output features
3 (no, unknown, yes)
Save Impulse
Add a processing block
Add a learning block

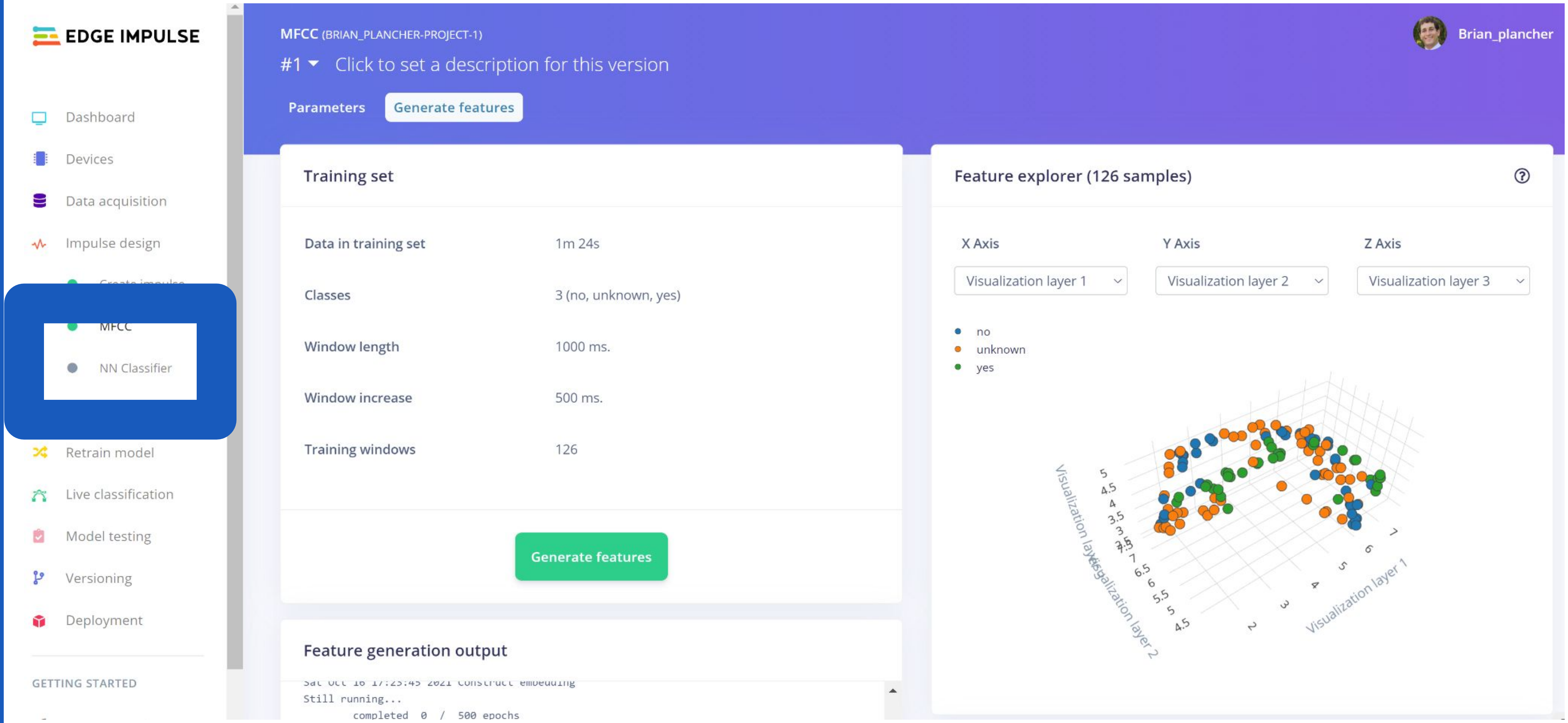

EDGE IMPULSE
Dashboard
Devices
Impulse design
Create impulse
MFCC
NN Classifier
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
CREATE IMPULSE (BRIAN_PLANCHER-PROJECT-1)
Brian_plancher
Successfully stored impulse. Configure the signal processing and learning blocks in the navigation bar.
Time series data
Axes
audio
Window size
1000 ms.
Window increase
500 ms.
Frequency (Hz)
16000
Zero-pad data
Audio (MFCC)
Name
MFCC
Input axes
audio
Classification (Keras)
Name
NN Classifier
Input features
MFCC
Output features
3 (no, unknown, yes)
Output features
3 (no, unknown, yes)
Save Impulse

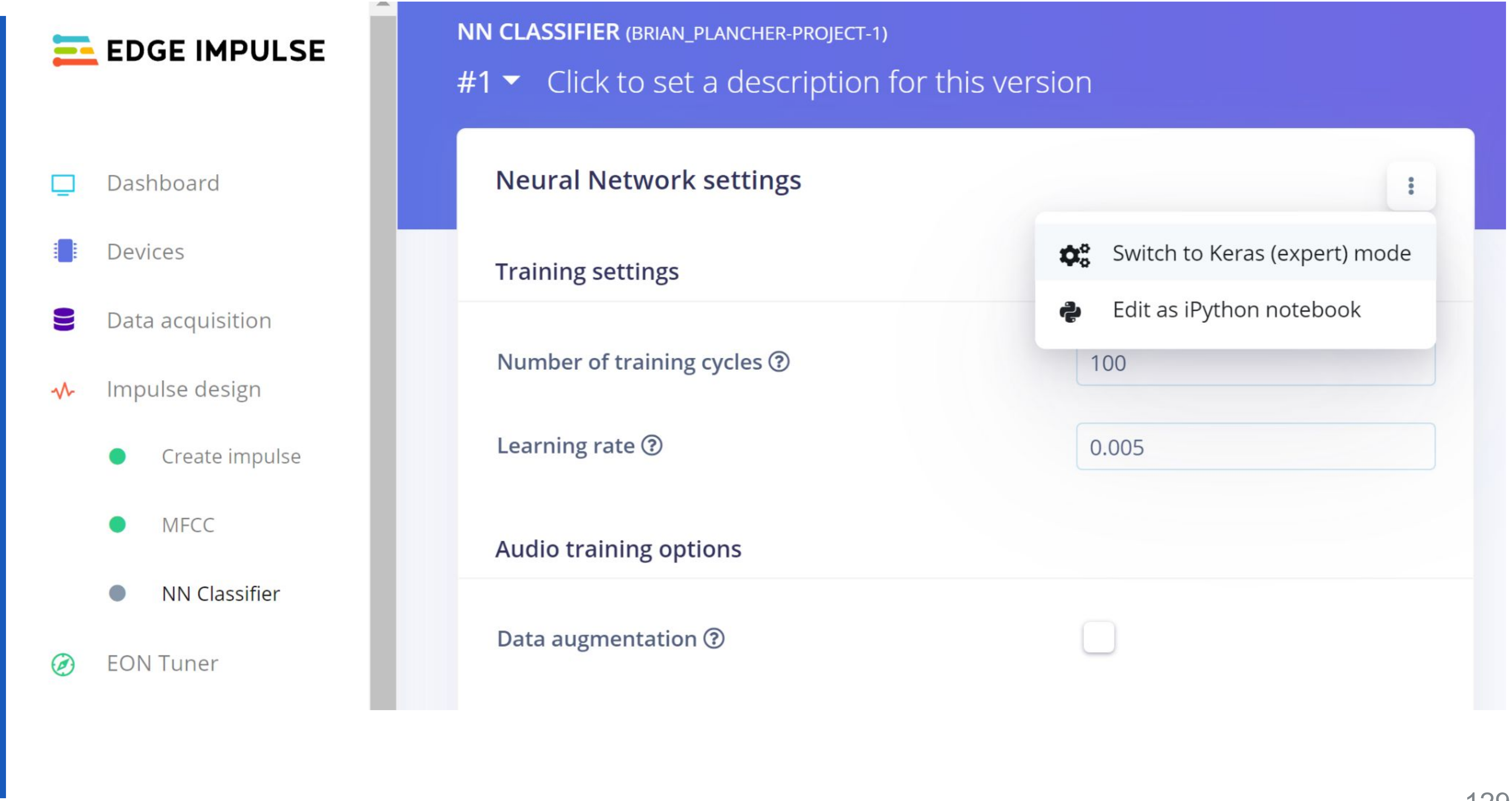

MFCC (BRIAN_PLANCHER-PROJECT-1)
Brian_plancher
#1
r this version
Paramet
Generate features
Raw data
0:00 / 0:01
unknown.2hvfrhdt (unknown)
20000
10000
0
-10000
-20000
audio
0
200
401
601
802
1002
1203
1403
1604
1804
Raw features
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,…
Parameters
Mel Frequency Cepstral Coefficients
Number of coefficients
13
Frame length
0.02
DSP result
Cepstral Coefficients
Processed features


MFCC (BRIAN_PLANCHER-PROJECT-1)
Brian_plancher
#1 Click to set a description for this version
Parameters
Generate features
Training set
Data in training set
1m 24s
Classes
3 (no, unknown, yes)
Window length
1000 ms.
Window increase
500 ms.
Training windows
126
Generate features
Feature explorer
No features generated yet.


EDGE IMPULSE
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
MFCC
NN Classifier
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
MFCC (BRIAN_PLANCHER-PROJECT-1)
#1 Click to set a description for this version
Brian_plancher
Parameters
Generate features
Training set
Data in training set 1m 24s
Classes 3 (no, unknown, yes)
Window length 1000 ms.
Window increase 500 ms.
Training windows 126
Generate features
Feature generation output
Still running...
completed 0 / 500 epochs
Feature explorer (126 samples)
X Axis
Y Axis
Z Axis
Visualization layer 1
Visualization layer 2
Visualization layer 3
no
unknown
yes
Visualization layer 1
Visualization layer 2

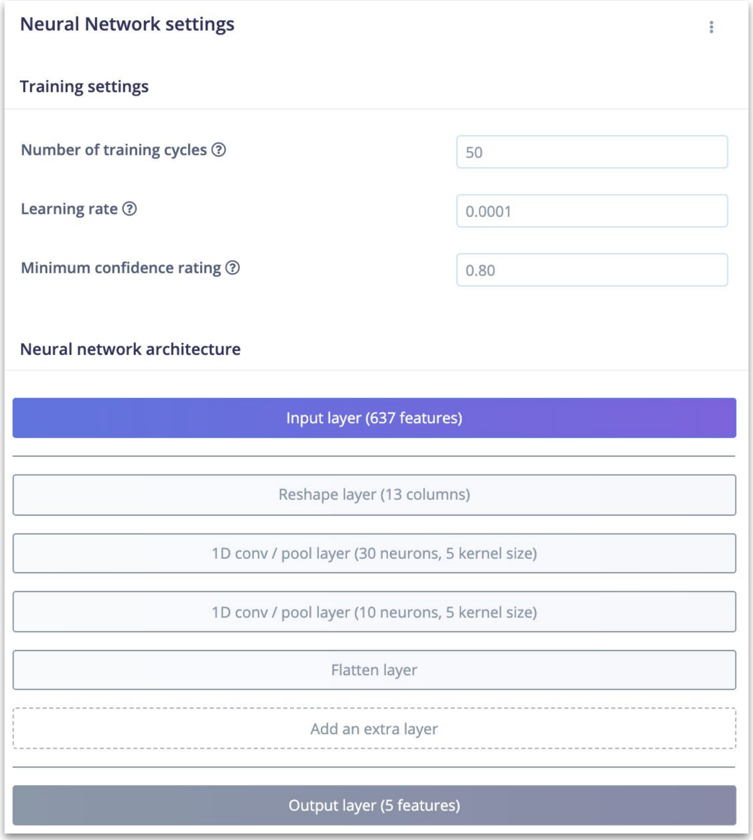

EDGE IMPULSE
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
MFCC
NN Classifier
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
MFCC (BRIAN_PLANCHER-PROJECT-1)
#1 Click to set a description for this version
Parameters
Generate features
Training set
Data in training set 1m 24s
Classes 3 (no, unknown, yes)
Window length 1000 ms.
Window increase 500 ms.
Training windows 126
Generate features
Feature generation output
Still running...
completed 0 / 500 epochs
Brian_plancher
Feature explorer (1,506 samples)
X Axis
Y Axis
Z Axis
accX RMS
accY RMS
accZ RMS
idle
updown
walk
accZ RMS
accY RMS
accX RMS
updown.9.1cjh52qu
Window: 4608 - 6608 ms.
Label: updown
View features
accX
accY
accZ


EDGE IMPULSE
Dashboard
Devices
Data acquisition
Impulse design
MFCC
NN Classifier
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
MFCC (BRIAN_PLANCHER-PROJECT-1)
Brian_plancher
#1 Click to set a description for this version
Parameters
Generate features
Training set
Data in training set 1m 24s
Classes 3 (no, unknown, yes)
Window length 1000 ms.
Window increase 500 ms.
Training windows 126
Generate features
Feature generation output
Still running...
completed 0 / 500 epochs
Feature explorer (126 samples)
X Axis
Y Axis
Z Axis
Visualization layer 1
Visualization layer 2
Visualization layer 3
no
unknown
yes


EDGE IMPULSE
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
MFCC
NN Classifier
EON Tuner
NN CLASSIFIER (BRIAN_PLANCHER-PROJECT-1)
#1
Click to set a description for this version
Neural Network settings
Switch to Keras (expert) mode
Edit as iPython notebook
Training settings
Number of training cycles
100
Learning rate
0.005
Audio training options
Data augmentation

Model Design with Edge Impulse

Pre-made neural network "blocks" that you can add!



Model Design with Edge Impulse

“**Expert**” mode to write

your own TensorFlow code

Neural network architecture

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, InputLayer, Dropout, Conv1D, Conv2D, Flatten, Reshape, MaxPooling1D, MaxPooling2D, BatchNormalization
from tensorflow.keras.optimizers import Adam
sys.path.append('./resources/libraries')
import ei_tensorflow.training

# model architecture
model = Sequential()
channels = 1
columns = 13
rows = int(input_length / (columns * channels))
model.add(Reshape((rows, columns, channels), input_shape=(input_length, )))
model.add(Conv2D(8, kernel_size=3, activation='relu', kernel_constraint=tf.keras.constraints.MaxNorm(1), padding='same'))
model.add(MaxPooling2D(pool_size=2, strides=2, padding='same'))
model.add(Dropout(0.25))
model.add(Conv2D(16, kernel_size=3, activation='relu', kernel_constraint=tf.keras.constraints.MaxNorm(1), padding='same'))
model.add(MaxPooling2D(pool_size=2, strides=2, padding='same'))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(classes, activation='softmax', name='y_pred'))
```

Start training

Neural network architecture

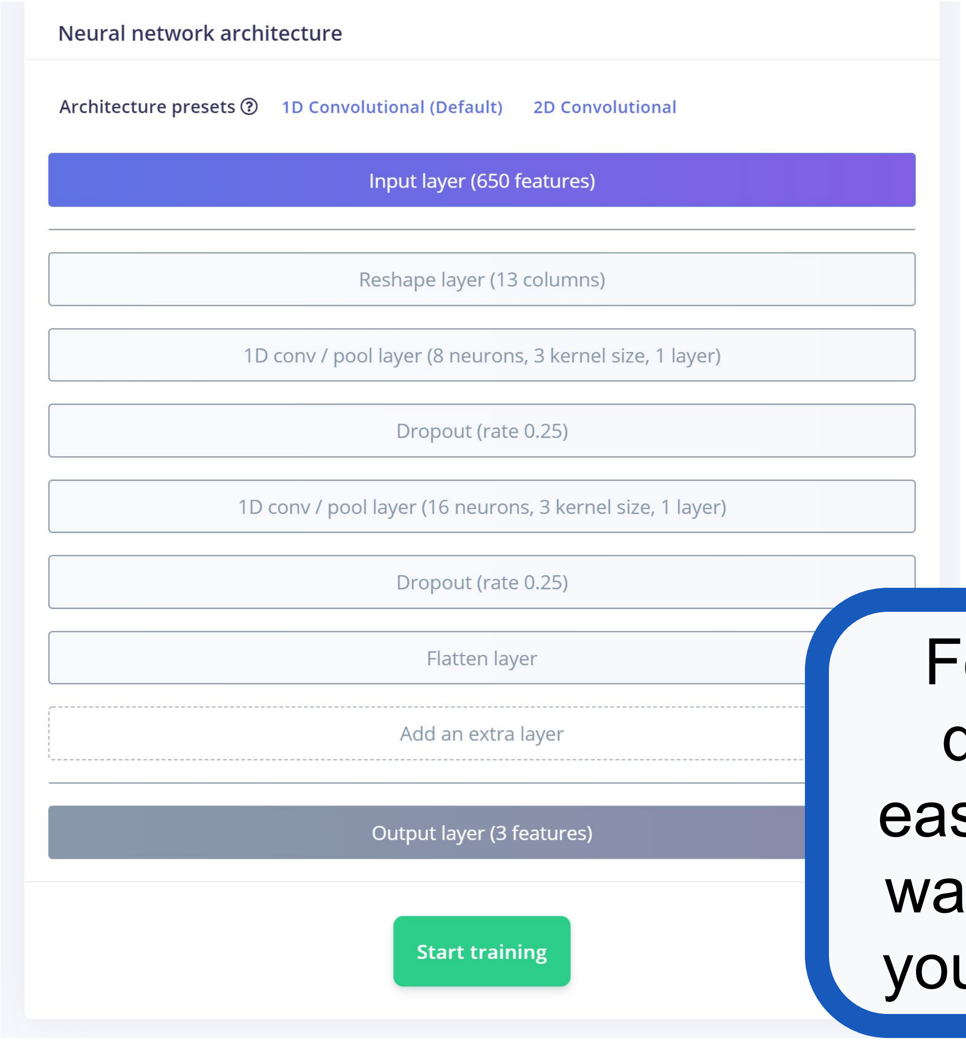
Architecture presets ⓘ 1D Convolutional (Default) 2D Convolutional

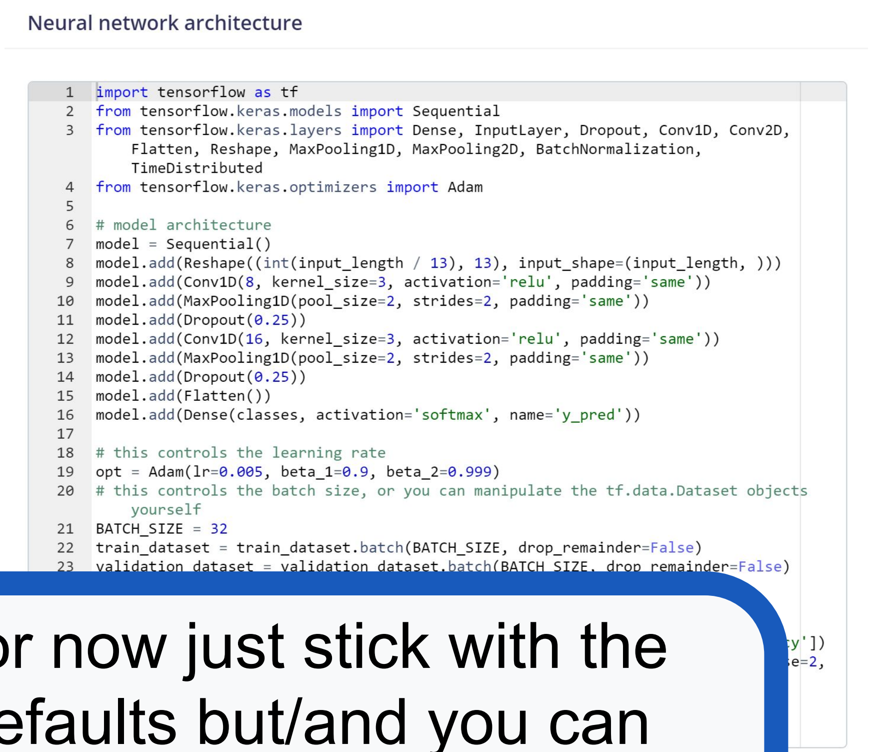
- Input layer (650 features)
- Reshape layer (13 columns)
- 1D conv / pool layer (8 neurons, 3 kernel size, 1 layer)
- Dropout (rate 0.25)
- 1D conv / pool layer (16 neurons, 3 kernel size, 1 layer)
- Dropout (rate 0.25)
- Flatten layer
- Add an extra layer
- Output layer (3 features)

Start training

Neural network architecture

```python
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, InputLayer, Dropout, Conv1D, Conv2D,
    Flatten, Reshape, MaxPooling1D, MaxPooling2D, BatchNormalization,
    TimeDistributed
from tensorflow.keras.optimizers import Adam

# model architecture

model.add(Reshape((int(input_length / 13), 13), input_shape=(input_length, )))
model.add(Conv1D(8, kernel_size=3, activation='relu', padding='same'))
model.add(MaxPooling1D(pool_size=2, strides=2, padding='same'))

model.add(Conv1D(16, kernel_size=3, activation='relu', padding='same'))
model.add(MaxPooling1D(pool_size=2, strides=2, padding='same'))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(classes, activation='softmax', name='y_pred'))

# this controls the learning rate
opt = Adam(lr=0.005, beta_1=0.9, beta_2=0.999)
# this controls the batch size, or you can manipulate the tf.data.Dataset objects
    yourself
BATCH_SIZE = 32
train_dataset = train_dataset.batch(BATCH_SIZE, drop_remainder=False)
validation_dataset = validation_dataset.batch(BATCH_SIZE, drop_remainder=False)
callbacks.append(BatchLoggerCallback(BATCH_SIZE, train_sample_count))

# train the neural network
model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy'])
model.fit(train_dataset, epochs=100, validation_data=validation_dataset, verbose=2,
    callbacks=callbacks)
```

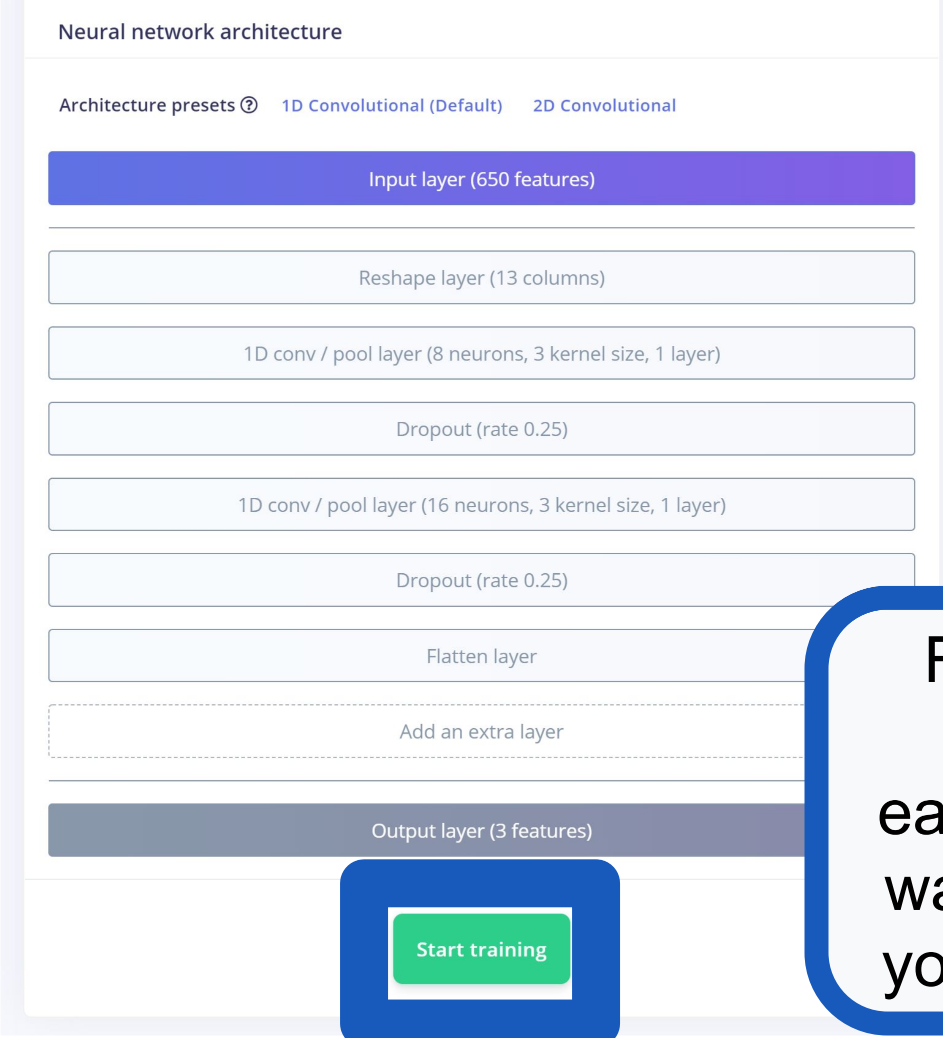




For now just stick with the defaults but/and you can easily design **any model** you want and use **any optimizer** you want using **TensorFlow**!

WARNING: if you want to deploy to a microcontroller make sure you only use Ops supported by TensorFlow Lite Micro! https://github.com/tensorflow/tflite-micro/blob/main/tensorflow/lite/micro/all_ops_resolver.cc#L22

the
can
easily design **any model** you want and use **any optimizer** you want using **TensorFlow**!



Neural network architecture

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, InputLayer, Dropout, Conv1D, Conv2D,
    Flatten, Reshape, MaxPooling1D, MaxPooling2D, BatchNormalization,
    TimeDistributed
from tensorflow.keras.optimizers import Adam

# model architecture
model = Sequential()
model.add(Reshape((int(input_length / 13), 13), input_shape=(input_length, )))
model.add(Conv1D(8, kernel_size=3, activation='relu', padding='same'))
model.add(MaxPooling1D(pool_size=2, strides=2, padding='same'))
model.add(Dropout(0.25))
model.add(Conv1D(16, kernel_size=3, activation='relu', padding='same'))
model.add(MaxPooling1D(pool_size=2, strides=2, padding='same'))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(classes, activation='softmax', name='y_pred'))

# this controls the learning rate
opt = Adam(lr=0.005, beta_1=0.9, beta_2=0.999)
# this controls the batch size, or you can manipulate the tf.data.Dataset objects
    yourself
BATCH_SIZE = 32
train_dataset = train_dataset.batch(BATCH_SIZE, drop_remainder=False)
validation_dataset = validation_dataset.batch(BATCH_SIZE, drop_remainder=False)
```

For now just stick with the defaults but/and you can easily design **any model** you want and use **any optimizer** you want using **TensorFlow**!

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 Accuracy

Model

Model version: Quantized (int8)

Last training performance (validation set)

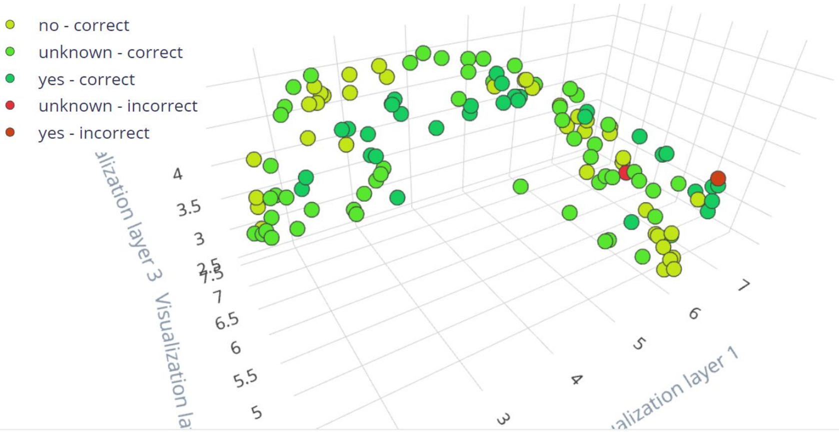
ACCURACY
92.3%

LOSS
0.27

Confusion matrix (validation set)

	NO	UNKNOWN	YES
NO	100%	0%	0%
UNKNOWN	9.1%	90.9%	0%
YES	0%	11.1%	88.9%
F1 SCORE	0.92	0.91	0.94

Feature explorer (full training set)



Final Accuracy

Accuracy Breakdown

Model

Model version: ? Quantized (int8)

Last training performance (validation set)

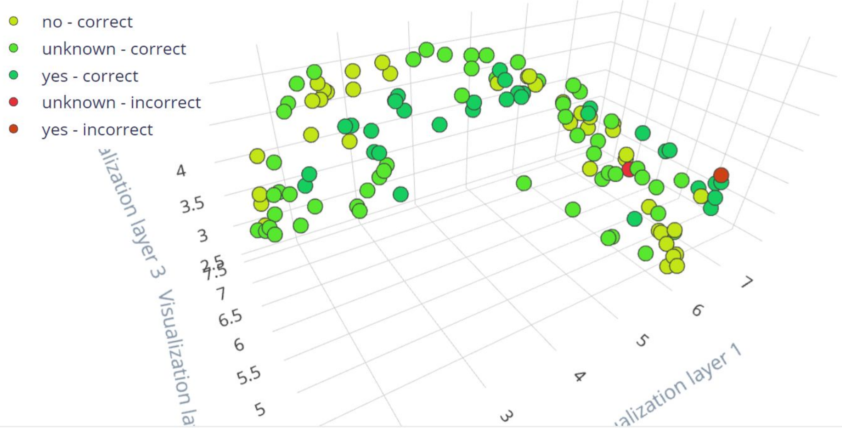
ACCURACY
92.3%

LOSS
0.27

Confusion matrix (validation set)

	NO	UNKNOWN	YES
NO	100%	0%	0%
UNKNOWN	9.1%	90.9%	0%
YES	0%	11.1%	88.9%
F1 SCORE	0.92	0.91	0.94

Feature explorer (full training set) ?



Confusion Matrix

	Actual Output = Yes	Actual Output = No
Predicted Output = Yes	**# of True Positive**	# of False Positive ***Type 1 Error***
Predicted Output = No	# of False Negative ***Type 2 Error***	**# of True Negative**

Final Accuracy

Accuracy Breakdown

Model

Model version: ? Quantized (int8)

Last training performance (validation set)

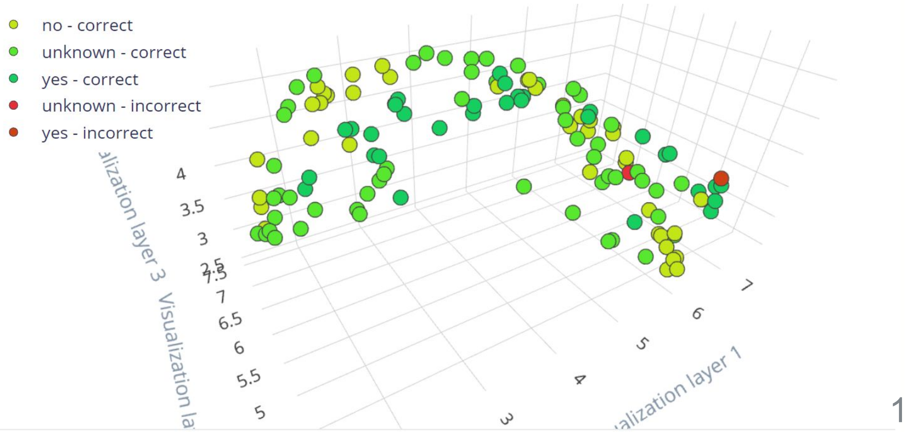
ACCURACY
92.3%

LOSS
0.27

Confusion matrix (validation set)

	NO	UNKNOWN	YES
NO	100%	0%	0%
UNKNOWN	9.1%	90.9%	0%
YES	0%	11.1%	88.9%
F1 SCORE	0.92	0.91	0.94

Feature explorer (full training set) ?



Final Accuracy

Accuracy Breakdown

Model

Model version: Quantized (int8)

Last training performance (validation set)

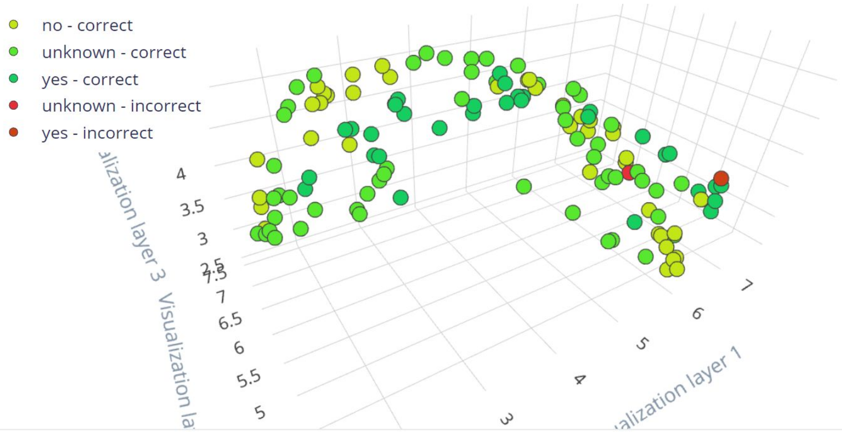
ACCURACY
92.3%

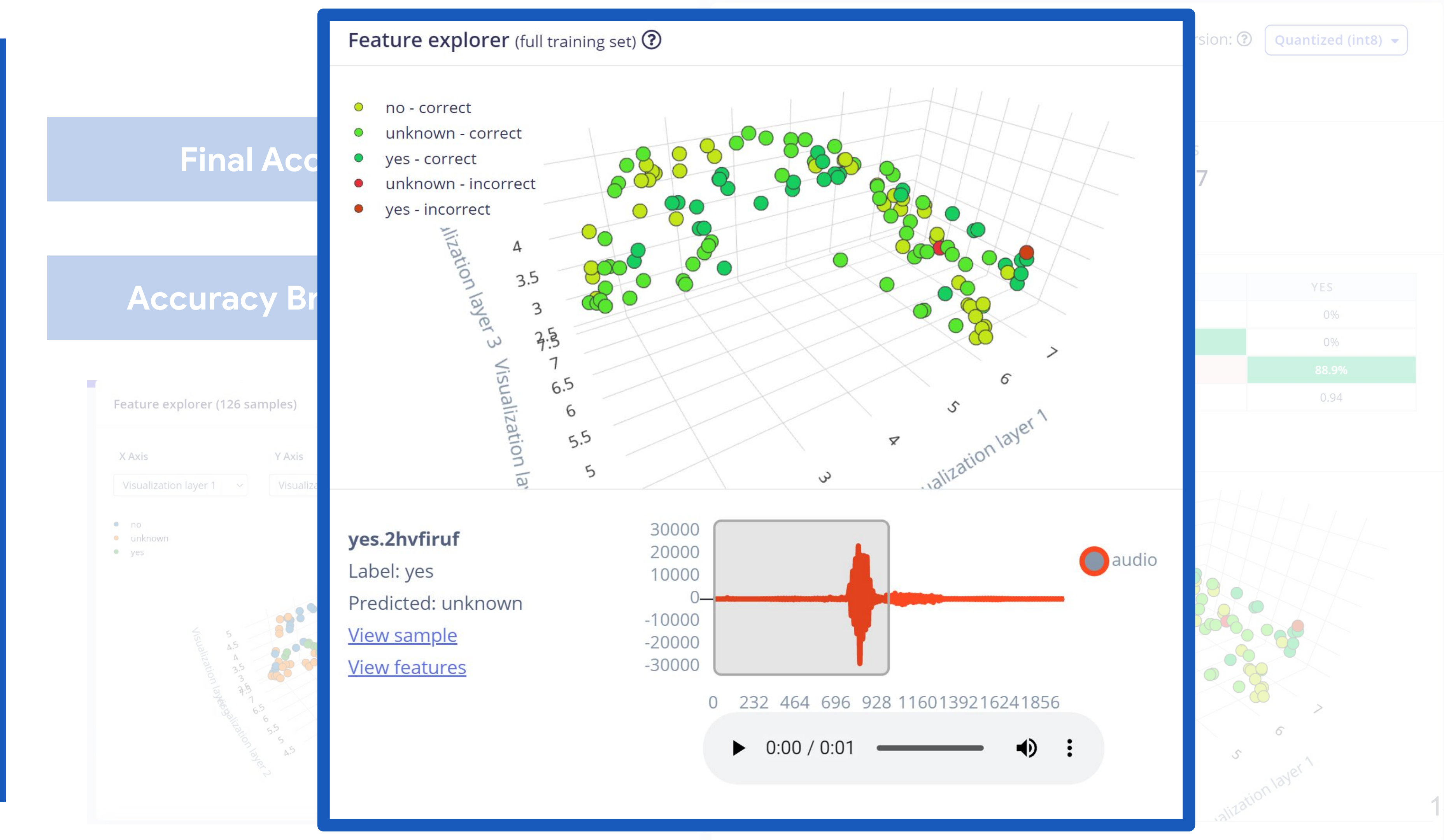
LOSS
0.27

Confusion matrix (validation set)

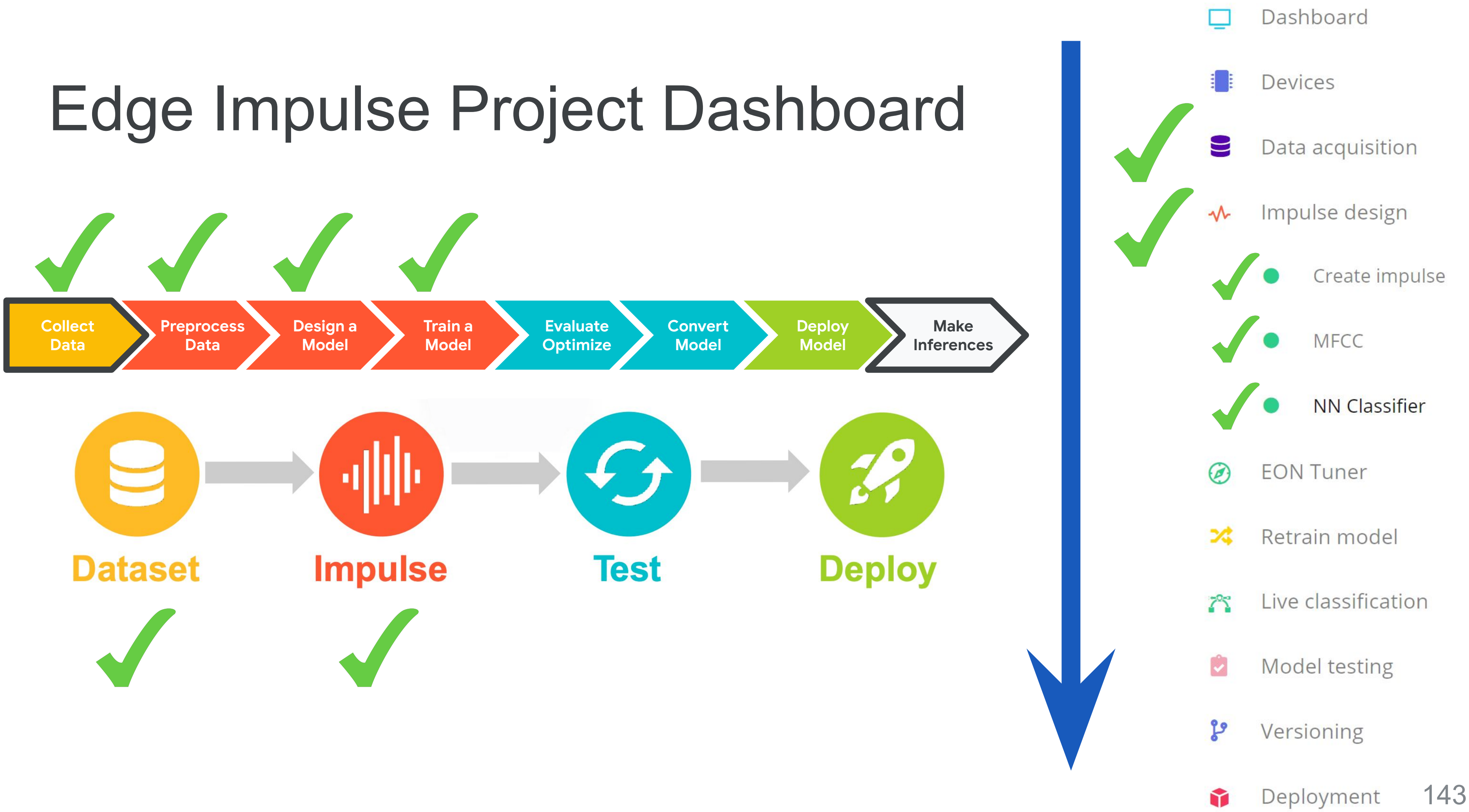
	NO	UNKNOWN	YES
NO	100%	0%	0%
UNKNOWN	9.1%	90.9%	0%
YES	0%	11.1%	88.9%
F1 SCORE	0.92	0.91	0.94

Feature explorer (full training set)




Feature explorer (full training set)
no - correct
unknown - correct
yes - correct
unknown - incorrect
yes - incorrect
Visualization layer 3
Visualization layer 1
yes.2hvfiruf
Label: yes
Predicted: unknown
View sample
View features
audio
0:00 / 0:01
Final Acc
Accuracy Br
Quantized (int8)
Feature explorer (126 samples)
X Axis
Y Axis
Visualization layer 1
YES
0%
0%
88.9%
0.94

Edge Impulse Project Dashboard



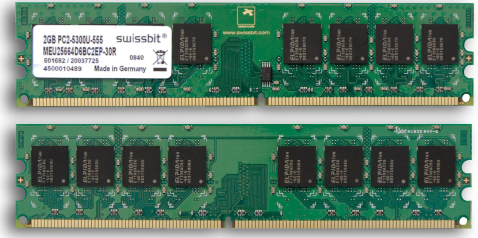
Today’s Agenda

- Deep ML Background
- Hands-on Computer Vision: Thing Translator
- The Tiny Machine Learning Workflow
- Keyword Spotting (KWS) Data Collection
- KWS Preprocessing and Training
- **Deployment Challenges and Opportunities for Embedded ML**
- Summary

Even Lower power
Even Lower bandwidth
Even Lower cost

Compute

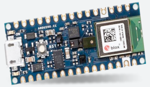
Memory



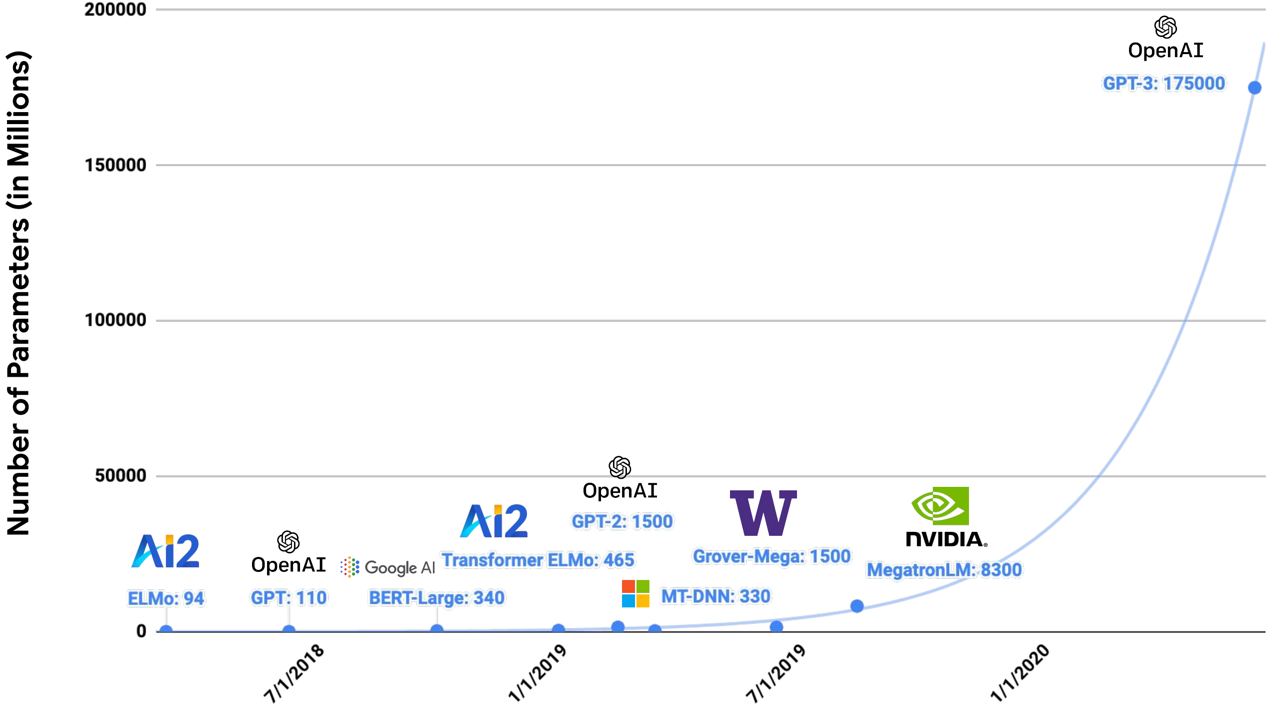
Storage

Microcontrollers have slower compute and very little memory and storage

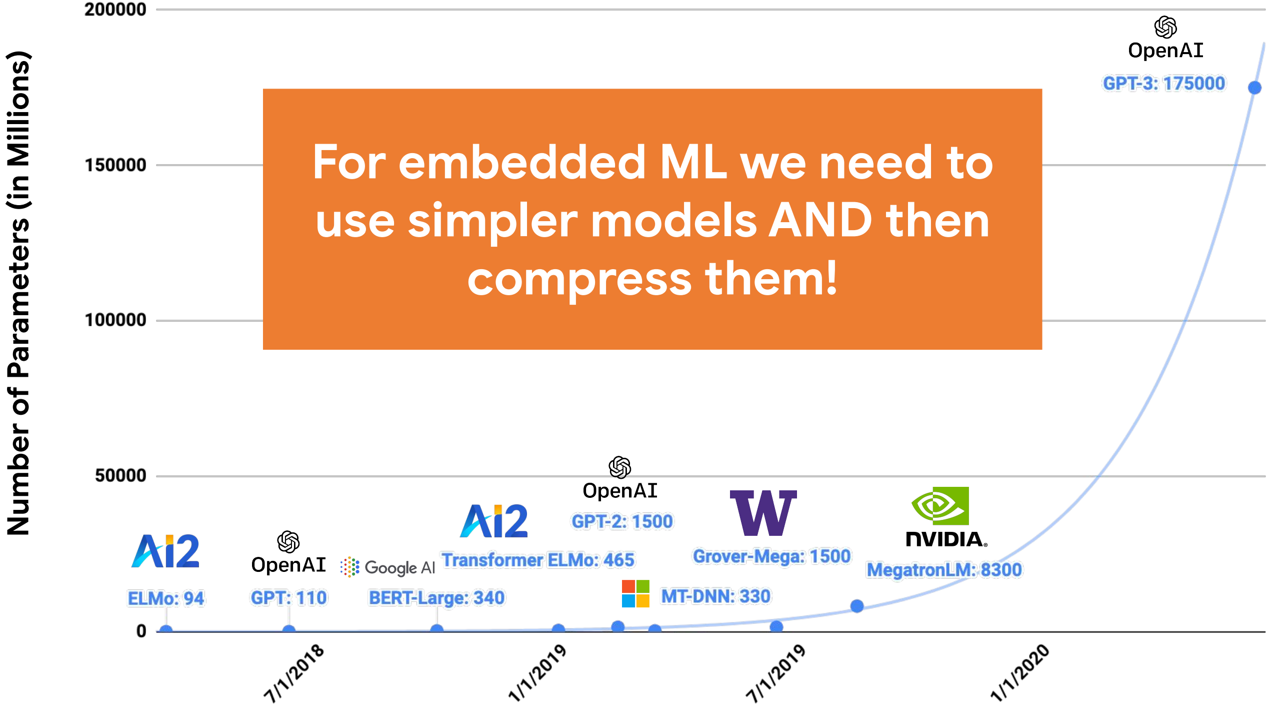
Orders of Magnitude Difference

	Computer 		Microcontroller 
Compute	1GHz–4GHz	~10X	1MHz–400MHz
Memory	512MB–64GB	~10,000X	2KB–512KB
Storage	64GB–4TB	~100,000X	32KB–2MB

ML Model Size Growth



ML Model Size Growth



For embedded ML we need to use simpler models AND then compress them!

Quantization

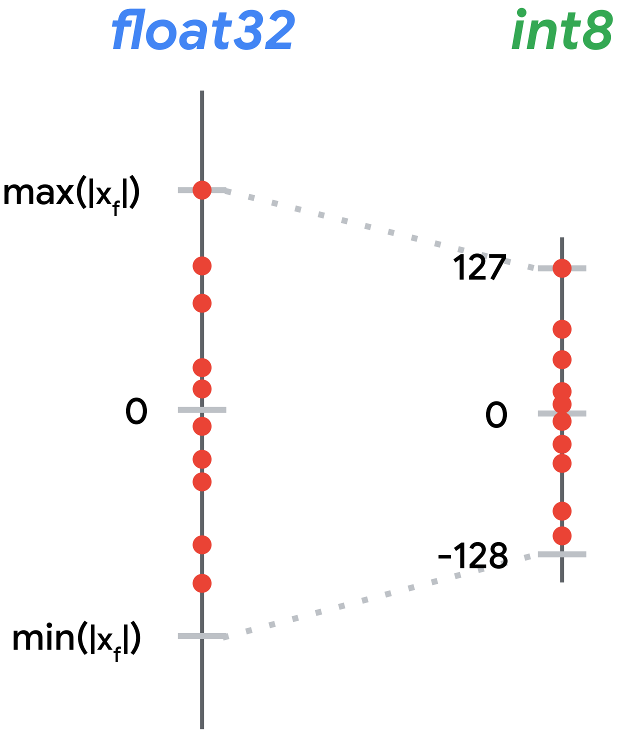
Reduces the precision of numbers used in a model which results in:

- 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%

Final Accuracy

Accuracy Breakdown

Model

Model version: Quantized (int8)

Last training performance (validation set)

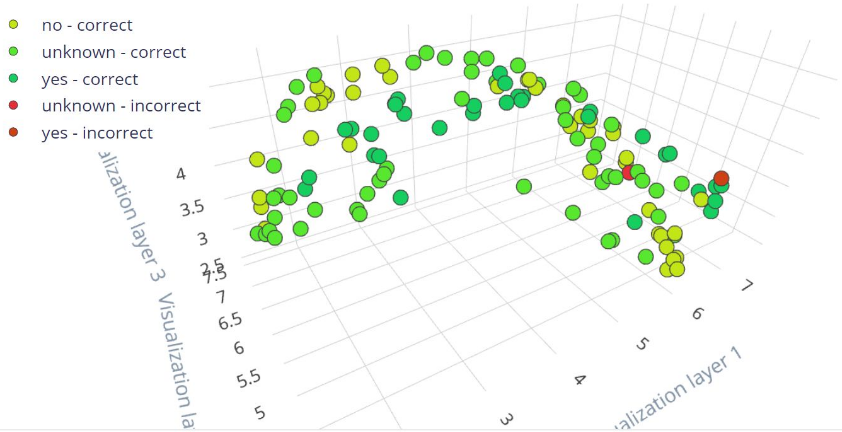
ACCURACY
92.3%

LOSS
0.27

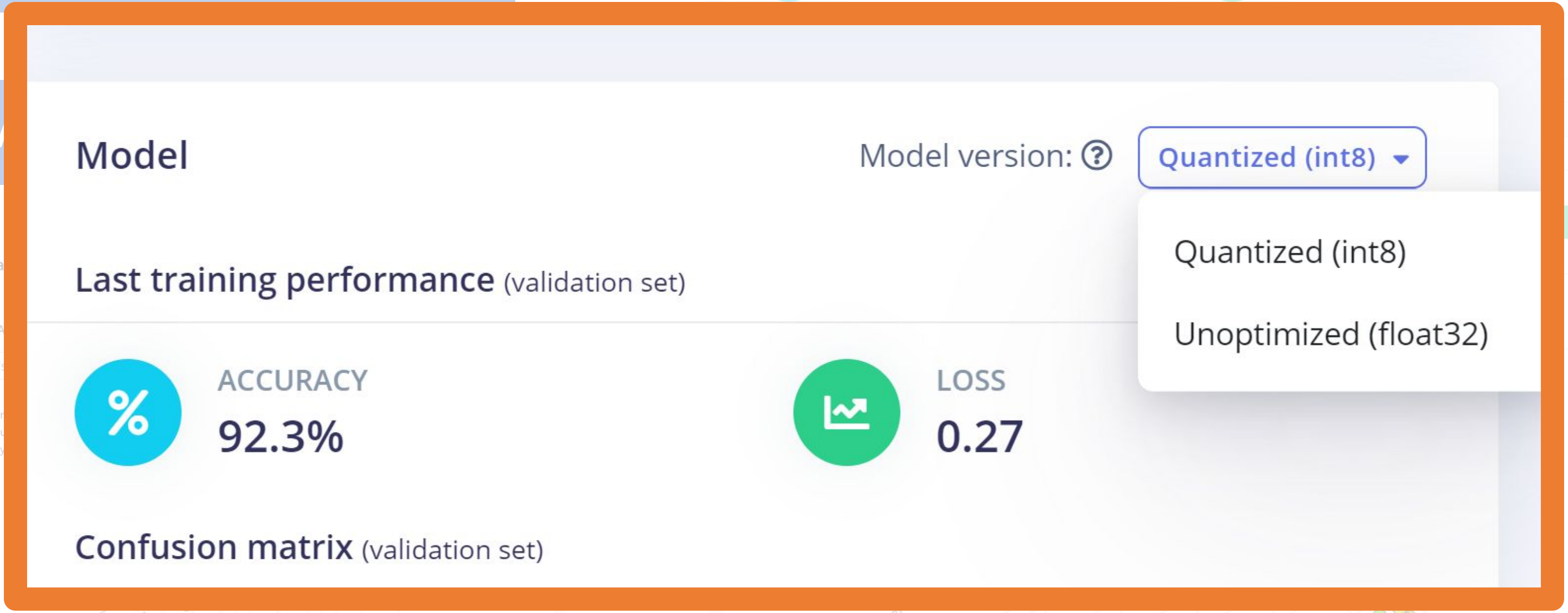
Confusion matrix (validation set)

	NO	UNKNOWN	YES
NO	100%	0%	0%
UNKNOWN	9.1%	90.9%	0%
YES	0%	11.1%	88.9%
F1 SCORE	0.92	0.91	0.94

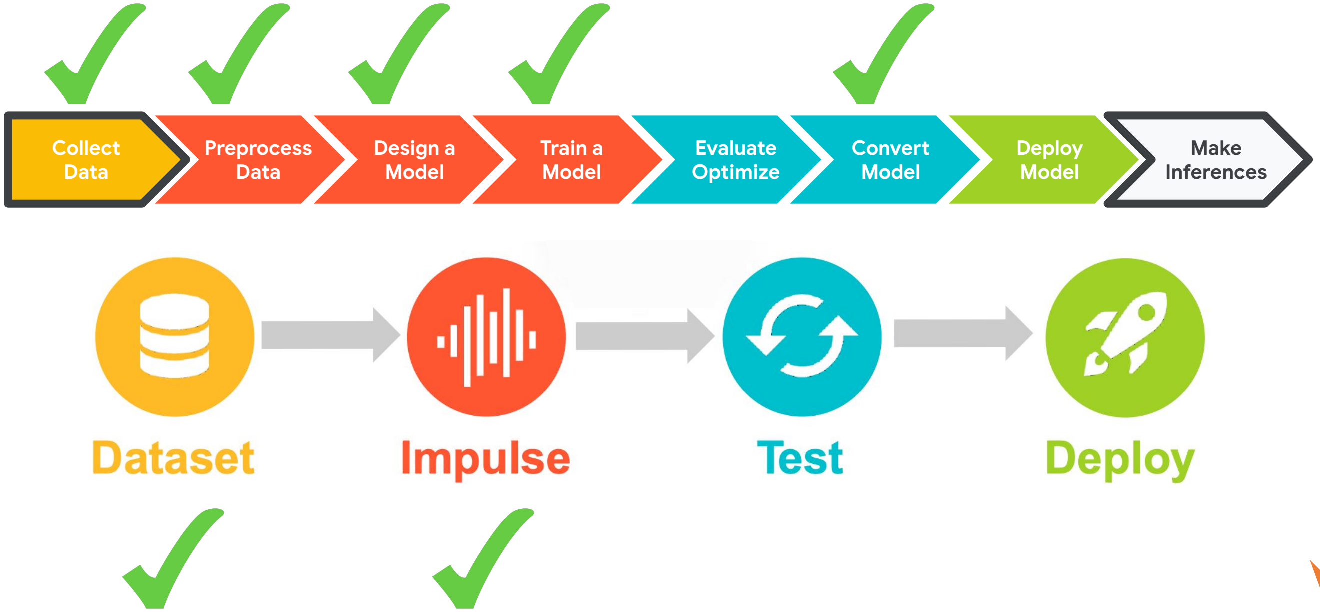
Feature explorer (full training set)

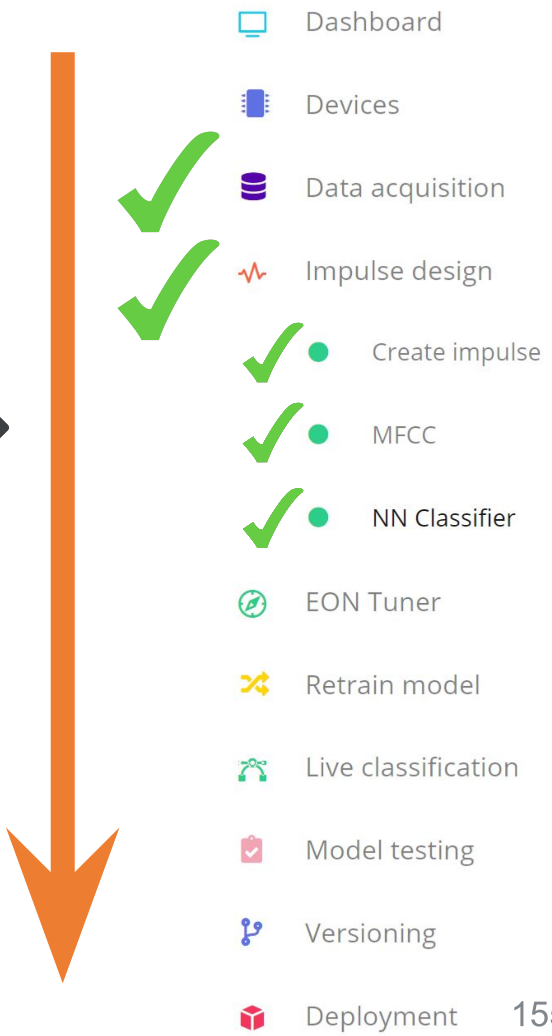


Final Accuracy

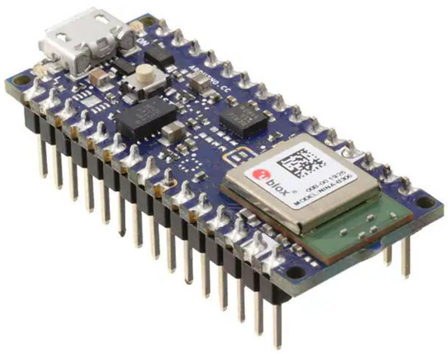


Edge Impulse Project Dashboard





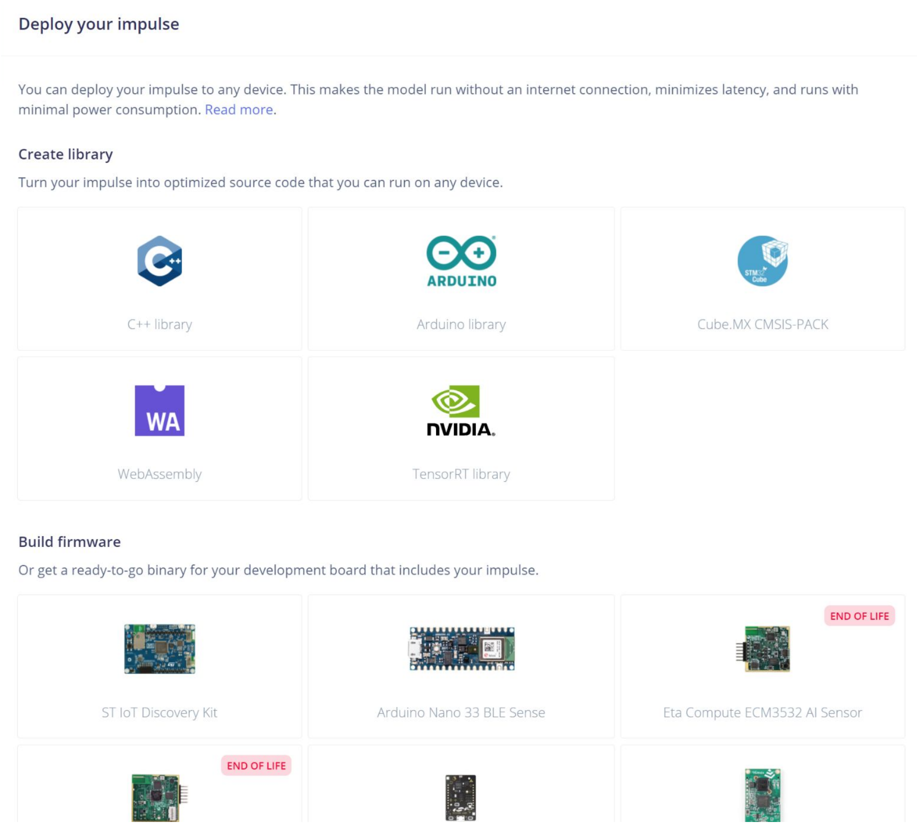
Most operating systems come with many libraries and applications that make it easy and portable to write code once and then compile it in an optimized form for most computers (or smartphones)



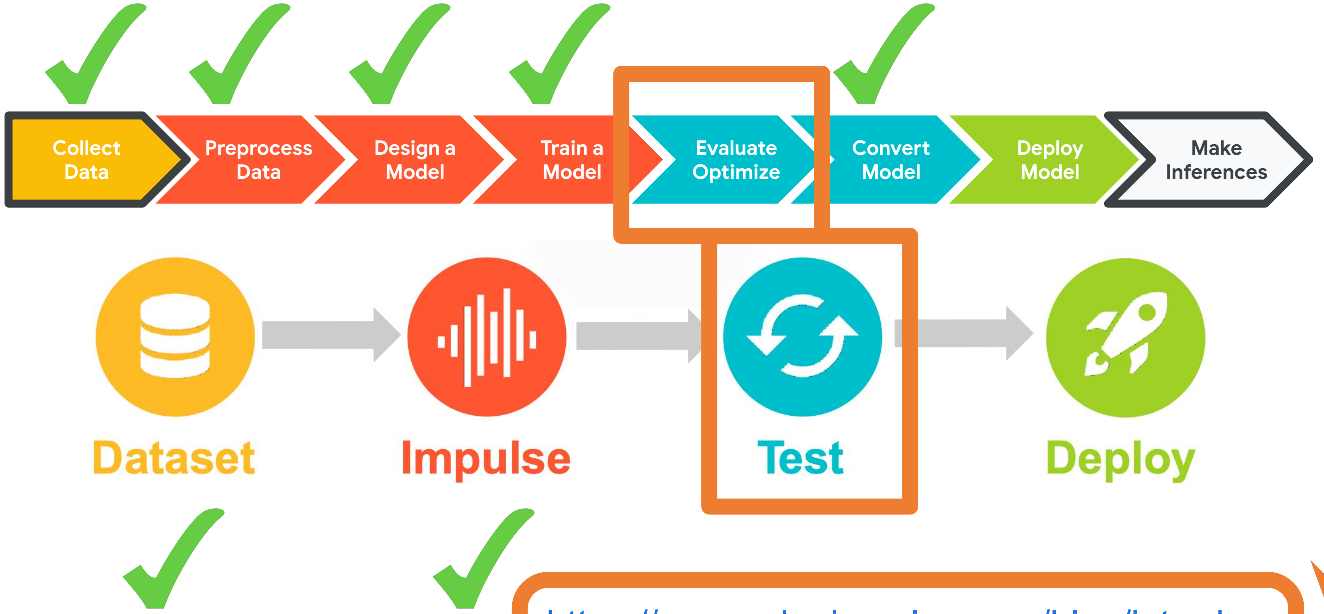
Microcontrollers often require custom code and compilation toolchains to run optimally

Edge Impulse simplifies deployment

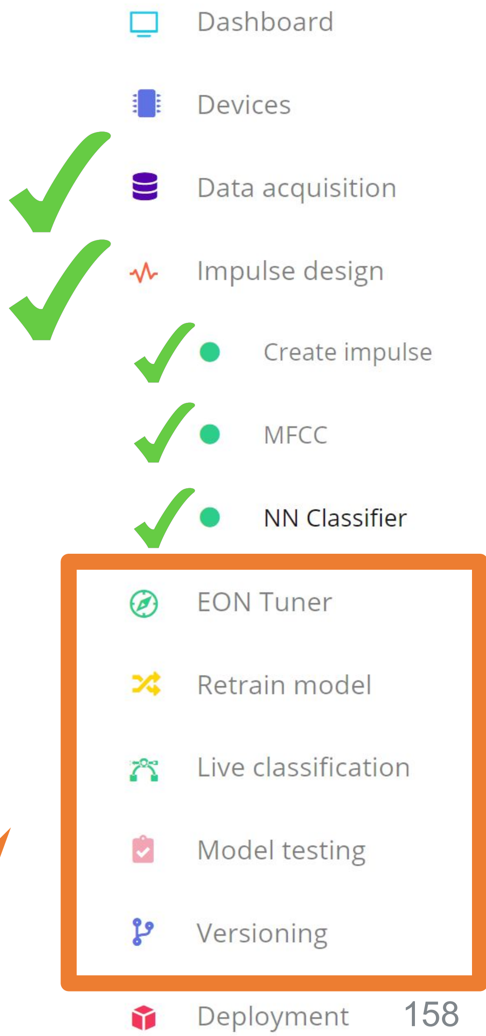
Pick your destination / device and **deploy** the same model to any of them thanks to **collaboration with hardware vendors** and the use of **TensorFlow Lite Micro**!



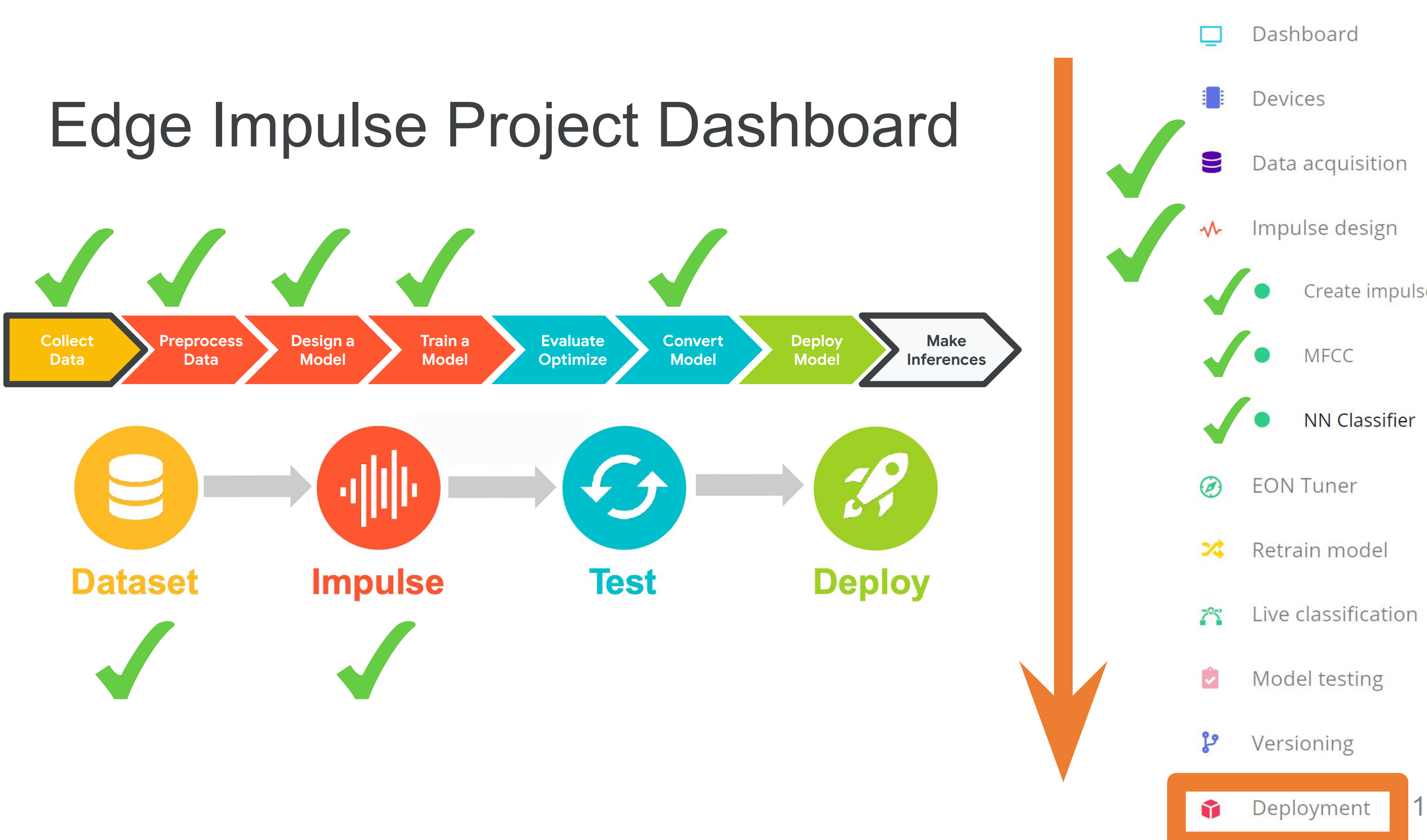
Edge Impulse Project Dashboard

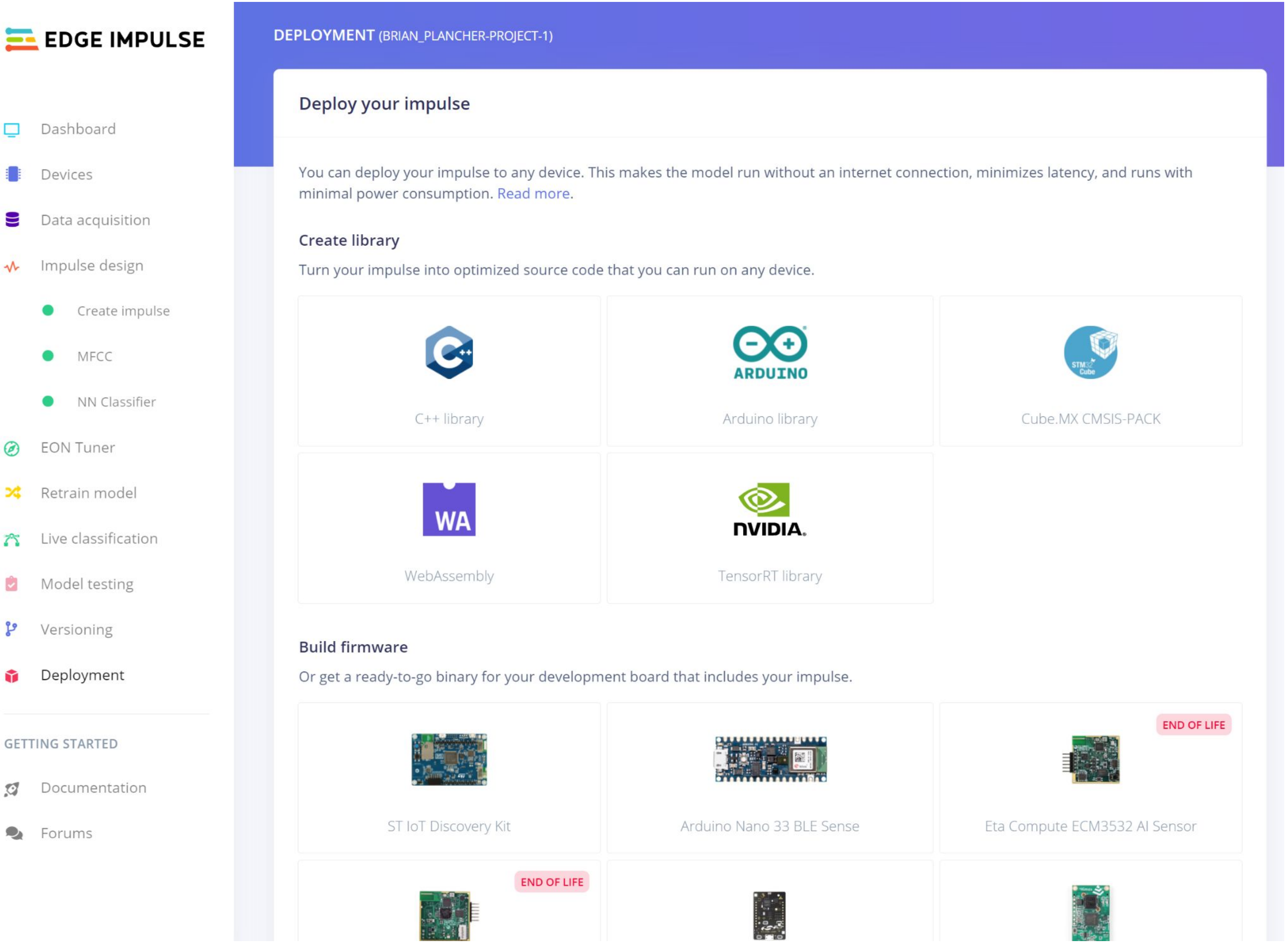


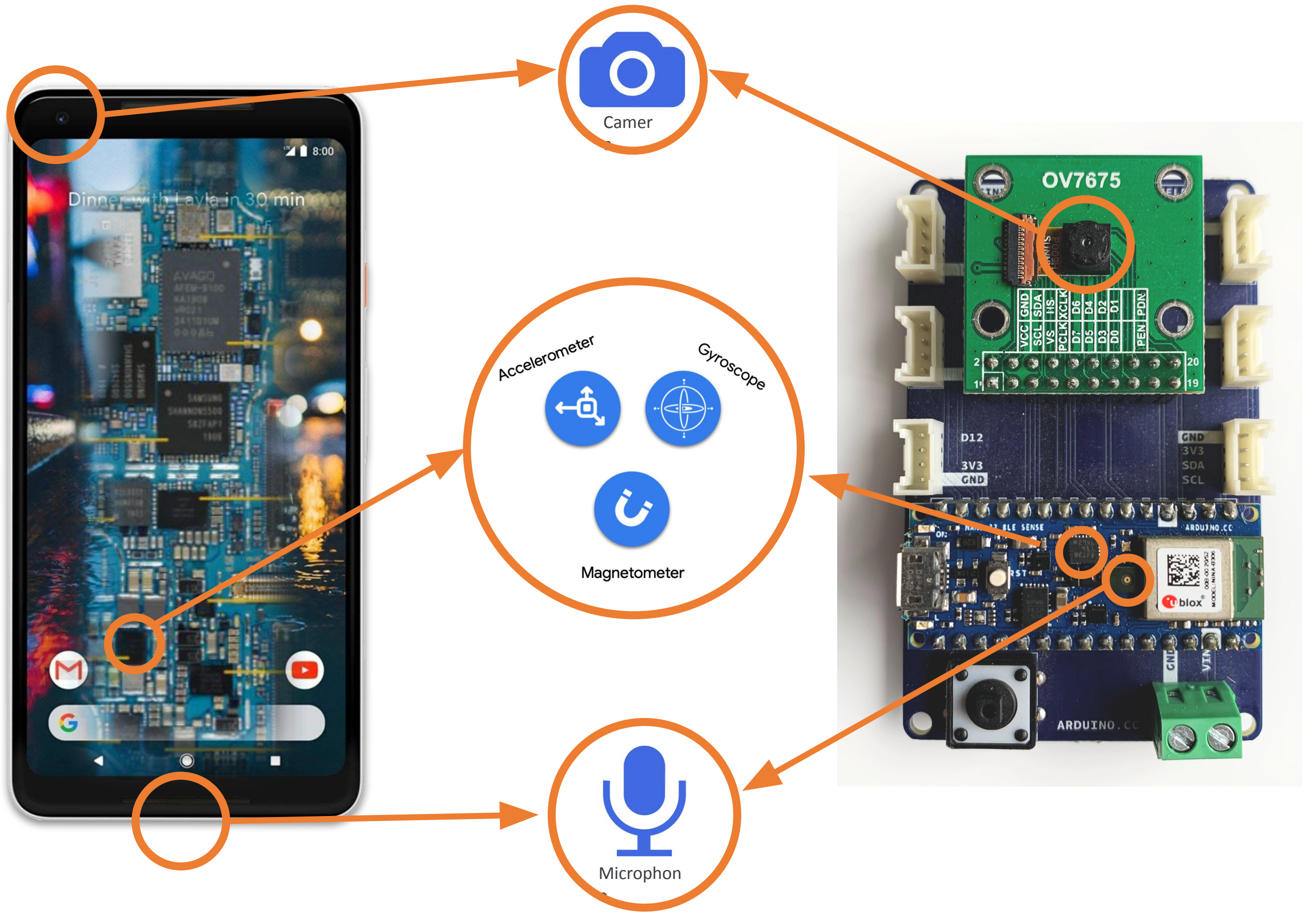
https://www.edgeimpulse.com/blog/introducing-the-eon-tuner-edge-impulses-new-automl-tool-for-embedded-machine-learning

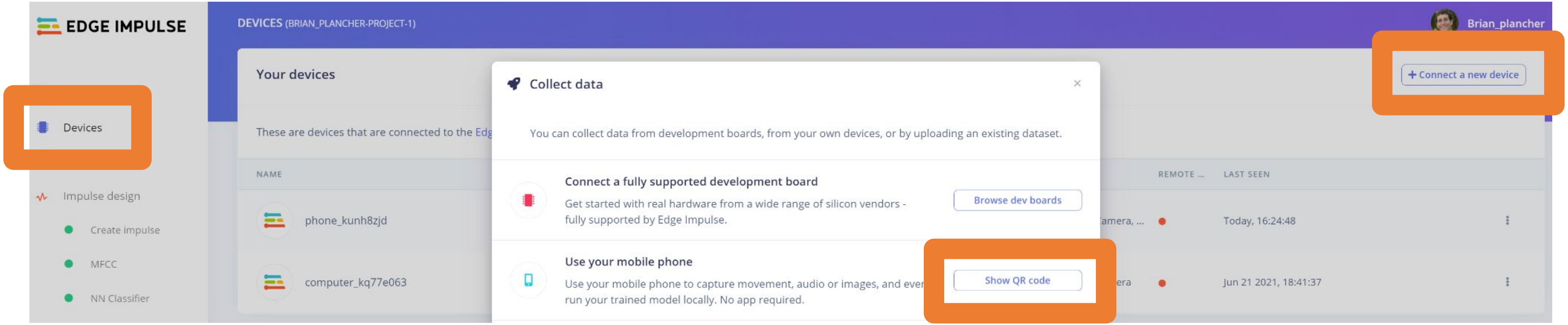


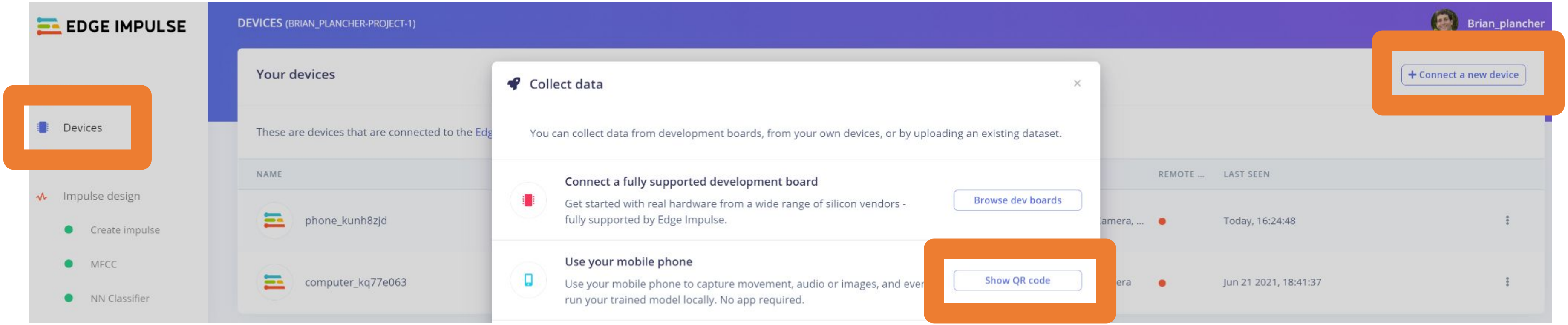
Edge Impulse Project Dashboard

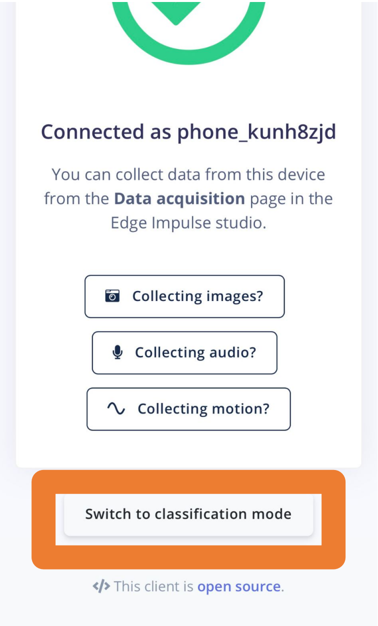


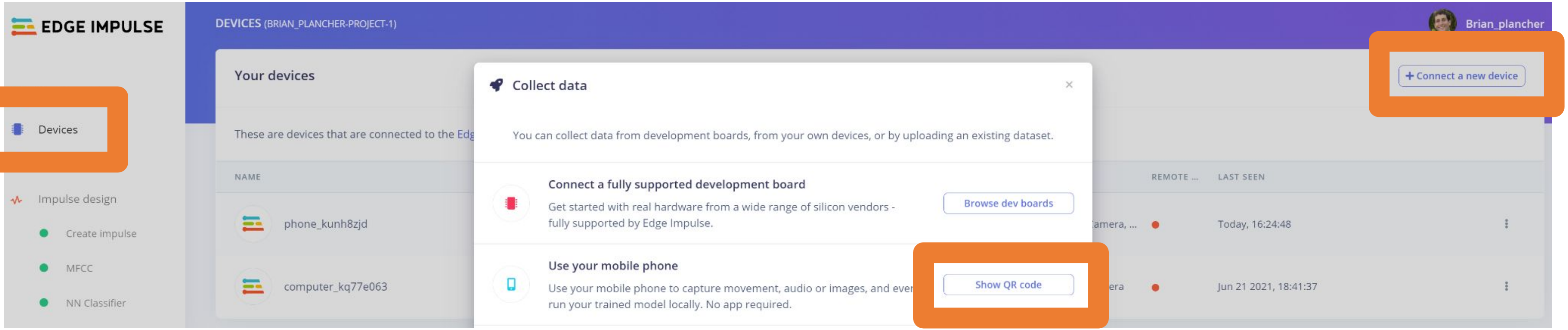

EDGE IMPULSE
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
MFCC
NN Classifier
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
DEPLOYMENT (BRIAN_PLANCHER-PROJECT-1)
Deploy your impulse
You can deploy your impulse to 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 source code that you can run on any device.
C++ library
ARDUINO
Arduino library
Cube.MX CMSIS-PACK
WA
WebAssembly
NVIDIA.
TensorRT library
Build firmware
Or get a ready-to-go binary for your development board that includes your impulse.
ST IoT Discovery Kit
Arduino Nano 33 BLE Sense
END OF LIFE
Eta Compute ECM3532 AI Sensor
END OF LIFE

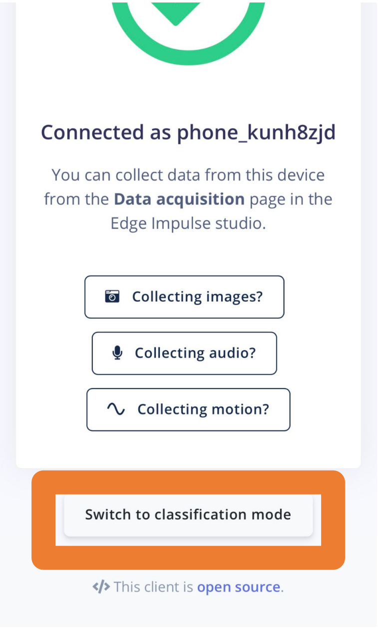

Camer
Accelerometer
Gyroscope
Magnetometer
Microphon
OV7675
ARDUINO.CC

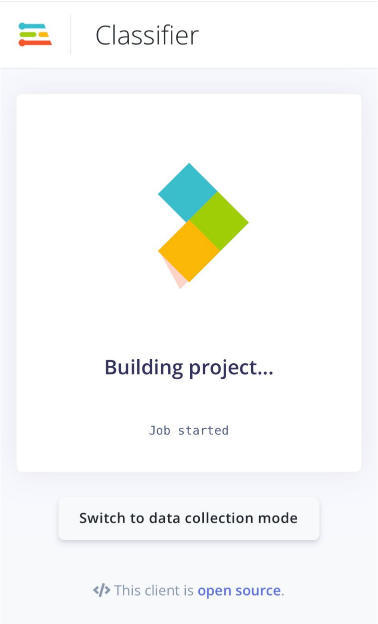

EDGE IMPULSE
DEVICES (BRIAN_PLANCHER-PROJECT-1)
Brian_plancher
Devices
Impulse design
Create impulse
MFCC
NN Classifier
Your devices
These are devices that are connected to the Edg
NAME
phone_kunh8zjd
computer_kq77e063
Collect data
You can collect data from development boards, from your own devices, 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.
Browse dev boards
Use your mobile phone
Use your mobile phone to capture movement, audio or images, and ever run your trained model locally. No app required.
Show QR code
+ Connect a new device
REMOTE ...
LAST SEEN
Today, 16:24:48
Jun 21 2021, 18:41:37

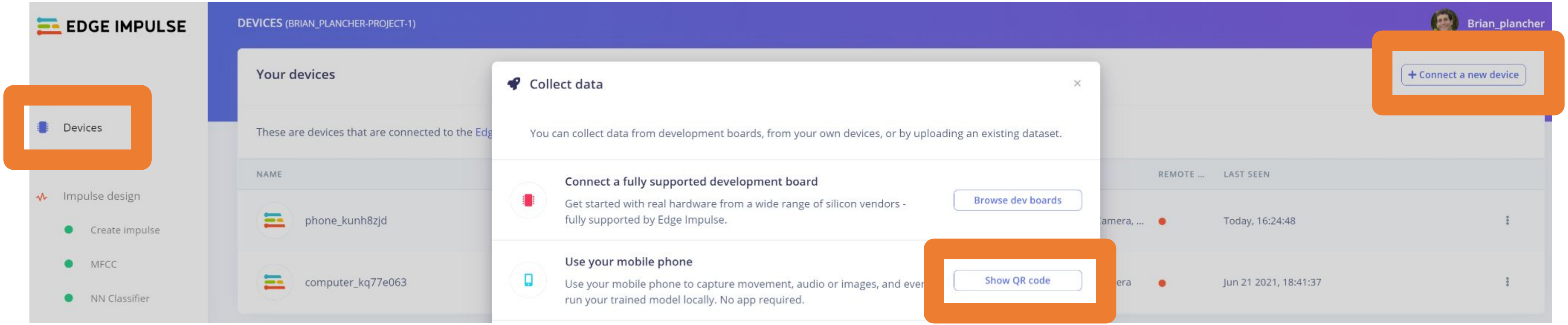

EDGE IMPULSE
DEVICES (BRIAN_PLANCHER-PROJECT-1)
Brian_plancher
Your devices
+ Connect a new device
Devices
These are devices that are connected to the Edg
Impulse design
Create impulse
MFCC
NN Classifier
NAME
phone_kunh8zjd
computer_kq77e063
Collect data
You can collect data from development boards, 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.
Browse dev boards
Use your mobile phone
Use your mobile phone to capture movement, audio or images, and ever run your trained model locally. No app required.
Show QR code
REMOTE ...
LAST SEEN
Today, 16:24:48
Jun 21 2021, 18:41:37

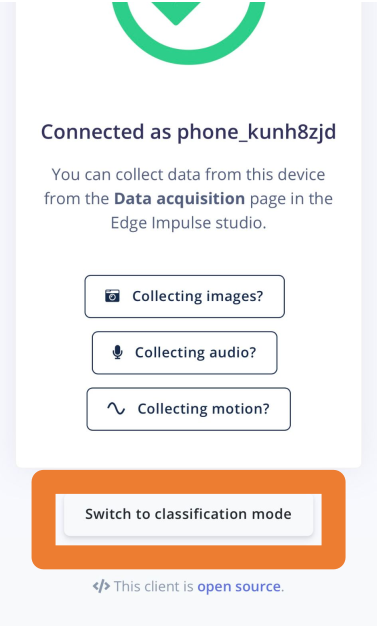

Connected as phone_kunh8zjd
You can collect data from this device from the Data acquisition page in the Edge Impulse studio.
Collecting images?
Collecting audio?
Collecting motion?
Switch to classification mode
This client is open source.

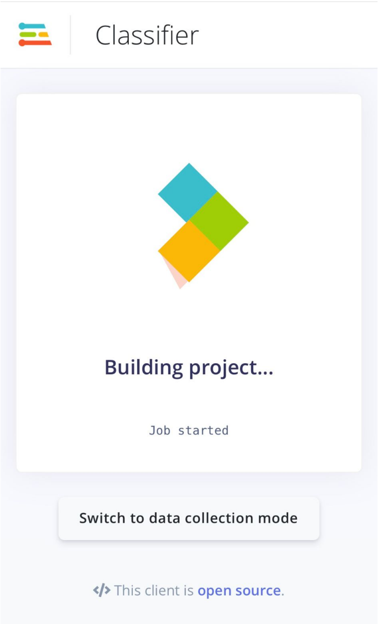

EDGE IMPULSE
DEVICES (BRIAN_PLANCHER-PROJECT-1)
Brian_plancher
Devices
Impulse design
Create impulse
MFCC
NN Classifier
Your devices
These are devices that are connected to the Edg
NAME
phone_kunh8zjd
computer_kq77e063
Collect data
You can collect data from development boards, from your own devices, 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.
Browse dev boards
Use your mobile phone
Use your mobile phone to capture movement, audio or images, and eve run your trained model locally. No app required.
Show QR code
+ Connect a new device
REMOTE ...
LAST SEEN
Today, 16:24:48
Jun 21 2021, 18:41:37


Connected as phone_kunh8zjd
You can collect data from this device from the Data acquisition page in the Edge Impulse studio.
Collecting images?
Collecting audio?
Collecting motion?
Switch to classification mode
This client is open source.


Classifier
Building project...
Job started
Switch to data collection mode
This client is open source.


EDGE IMPULSE
DEVICES (BRIAN_PLANCHER-PROJECT-1)
Brian_plancher
Devices
Impulse design
Create impulse
MFCC
NN Classifier
Your devices
These are devices that are connected to the Edg
NAME
phone_kunh8zjd
computer_kq77e063
+ Connect a new device
REMOTE ...
LAST SEEN
Today, 16:24:48
Jun 21 2021, 18:41:37
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.
Browse dev boards
Use your mobile phone
Use your mobile phone to capture movement, audio or images, and eve run your trained model locally. No app required.
Show QR code


Connected as phone_kunh8zjd
You can collect data from this device from the Data acquisition page in the Edge Impulse studio.
Collecting images?
Collecting audio?
Collecting motion?
Switch to classification mode
This client is open source.


Classifier
Building project...
Job started
Switch to data collection mode
This client is open source.


Waiting
yes
NO
UNKNOWN
YES
178 0.07 0.06 0.87
177 0.00 0.05 0.94
176 0.00 0.06 0.94
175 0.00 0.10 0.90

Deploy and Test your Model

Shows the **score** for **(confidence that the current sounds is)** each of the various keywords and unknown and bolds the highest score.

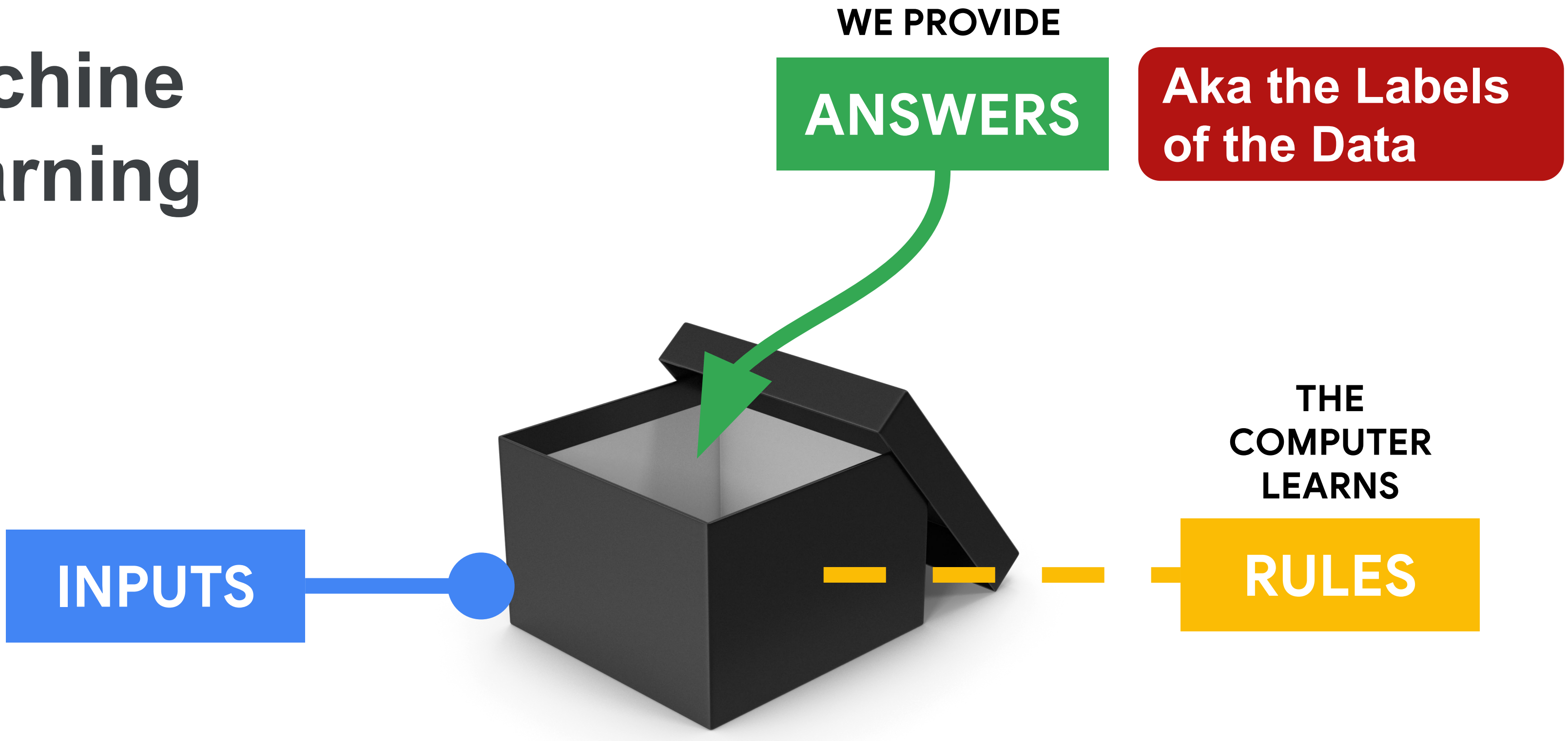
yes

	NO	UNKNOWN	YES
178	0.07	0.06	**0.87**
177	0.00	0.05	**0.94**
176	0.00	0.06	**0.94**
175	0.00	0.10	**0.90**

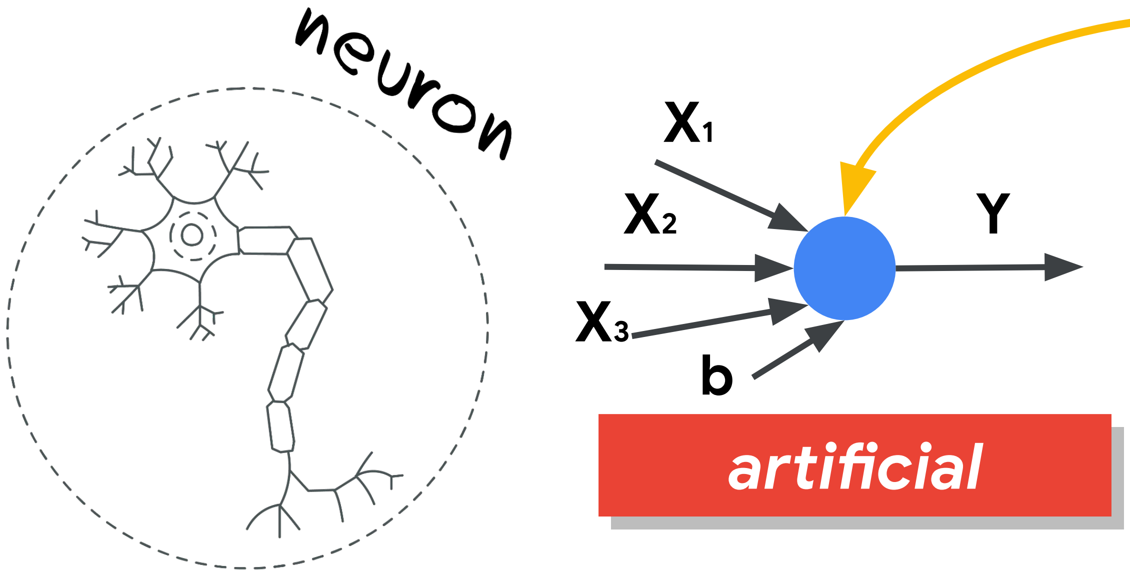
Today's Agenda

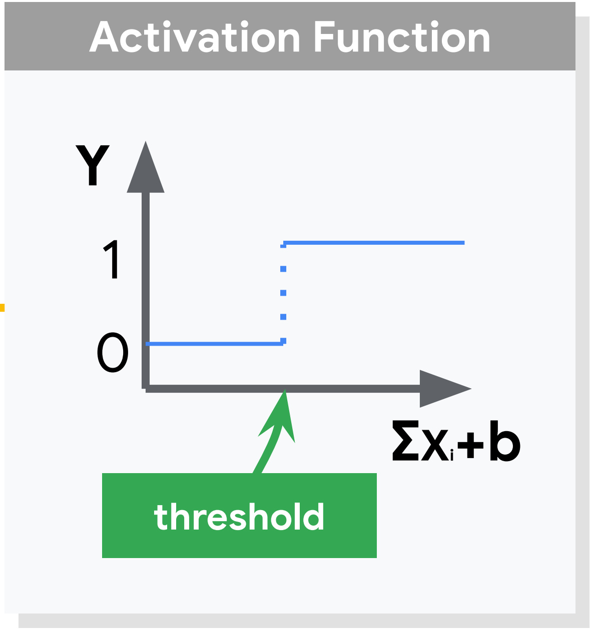
- Deep ML Background
- Hands-on Computer Vision: Thing Translator
- The Tiny Machine Learning Workflow
- Keyword Spotting (KWS) Data Collection
- KWS Preprocessing and Training
- Deployment Challenges and Opportunities for Embedded ML
- **Summary**

Machine Learning



What is a **neural network?**

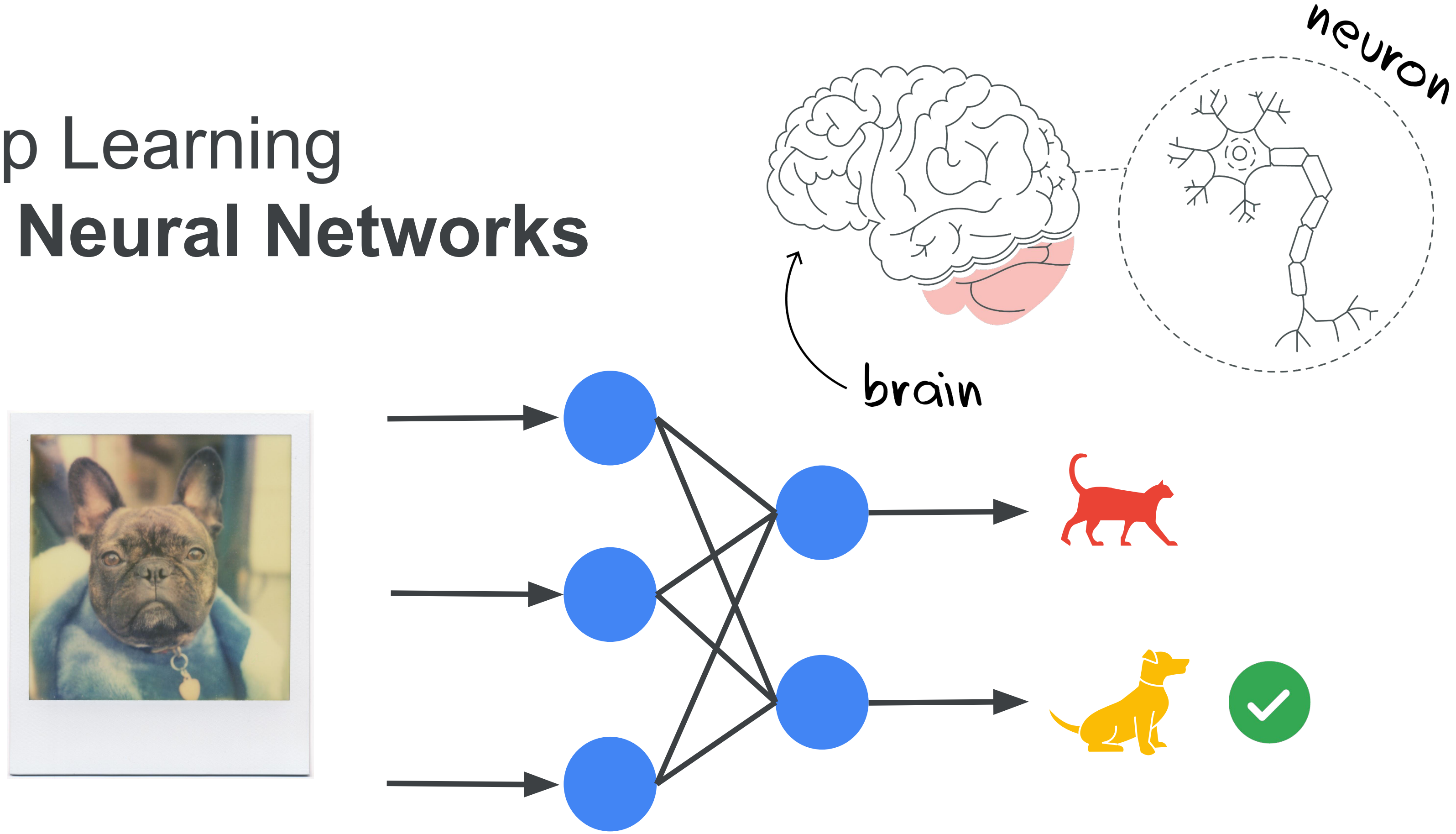




$$Y = \Sigma w_i x_i + b$$

So training the model is finding the right values for w_i and b

Deep Learning with **Neural Networks**

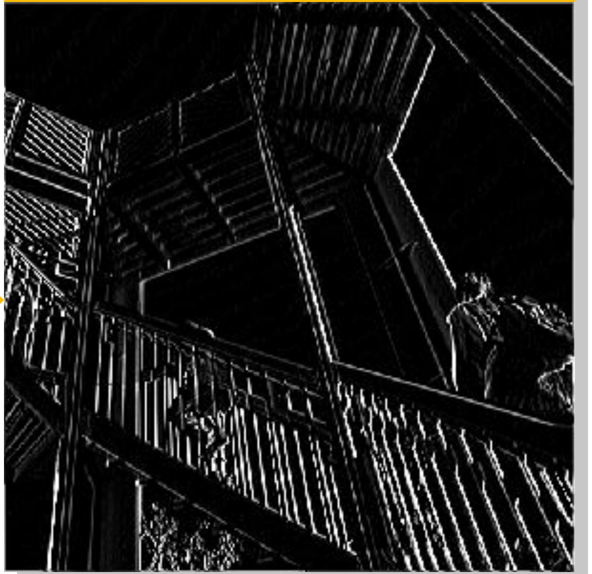


[Colab Link]

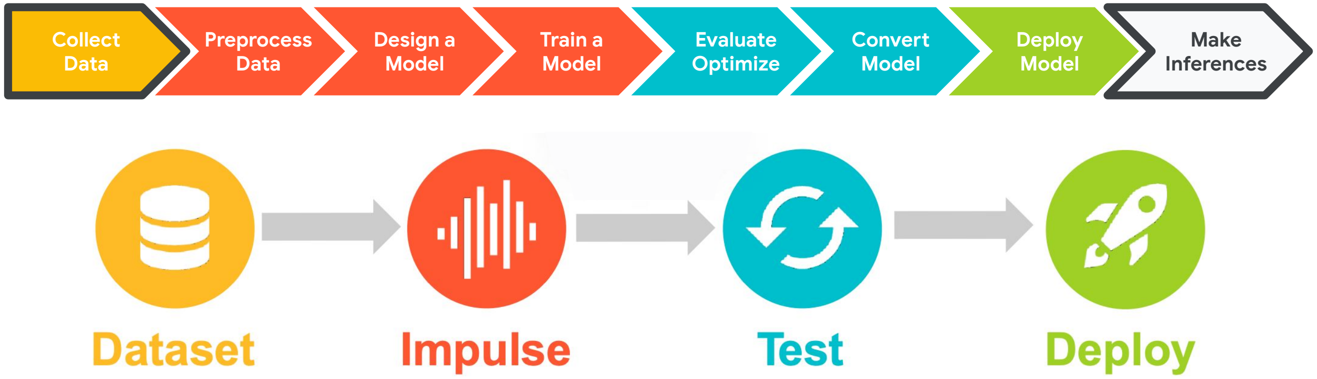
Features can be found with **Convolutions**

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

Features



The TinyML Workflow



The TinyML Workflow

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

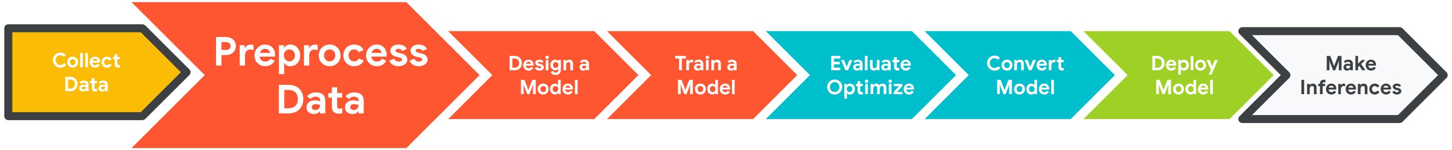
Who will use your ML model?

Where will your ML model be used?

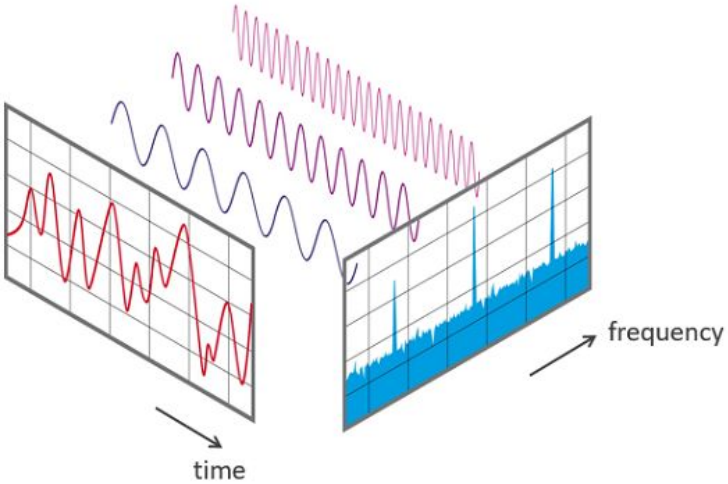
Why will your ML model be used?
Why those Keywords?

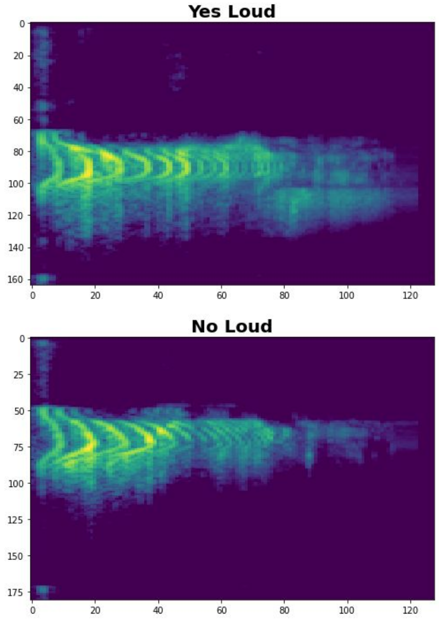
Training Set	Validation Set	Test Set

The TinyML Workflow

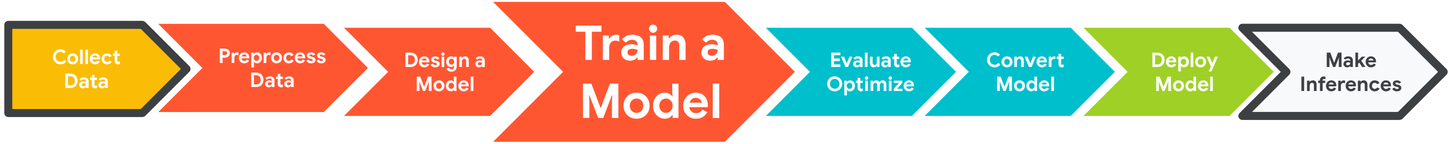


FFT, Spectrogram, MFCC





The **TinyML** Workflow



Confusion Matrix	Actual Output = Yes	Actual Output = No
Predicted Output = Yes	# of True Positive	# of False Positive ***Type 1 Error***
Predicted Output = No	# of False Negative ***Type 2 Error***	# of True Negative

The TinyML Workflow

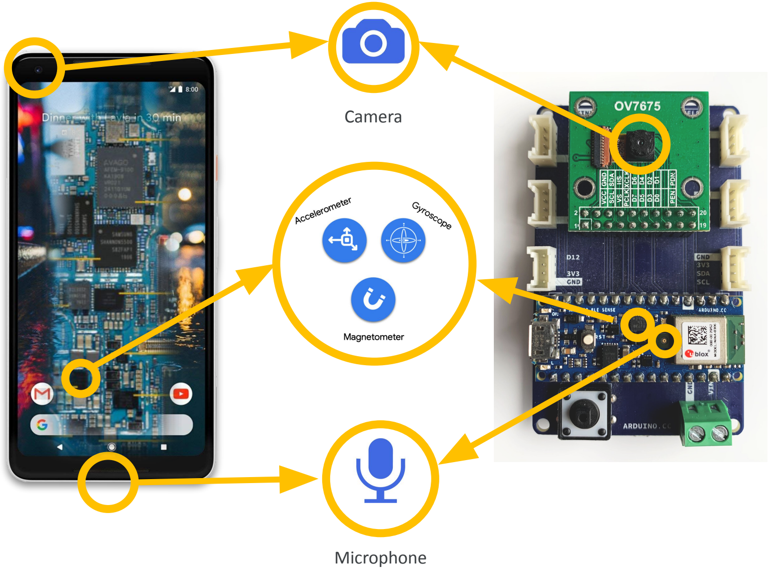
Quantization

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

- **smaller model size**
- **faster computation**

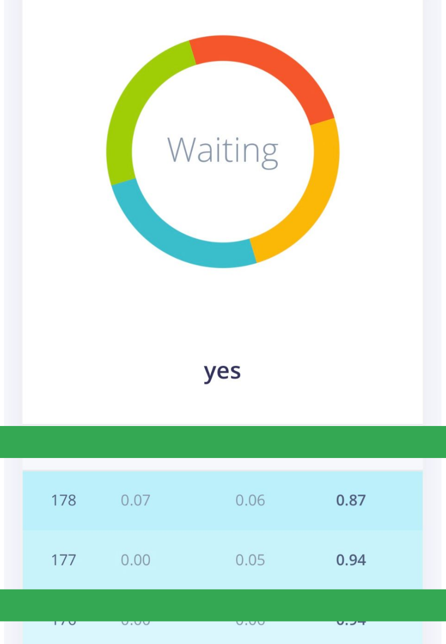
The TinyML Workflow

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

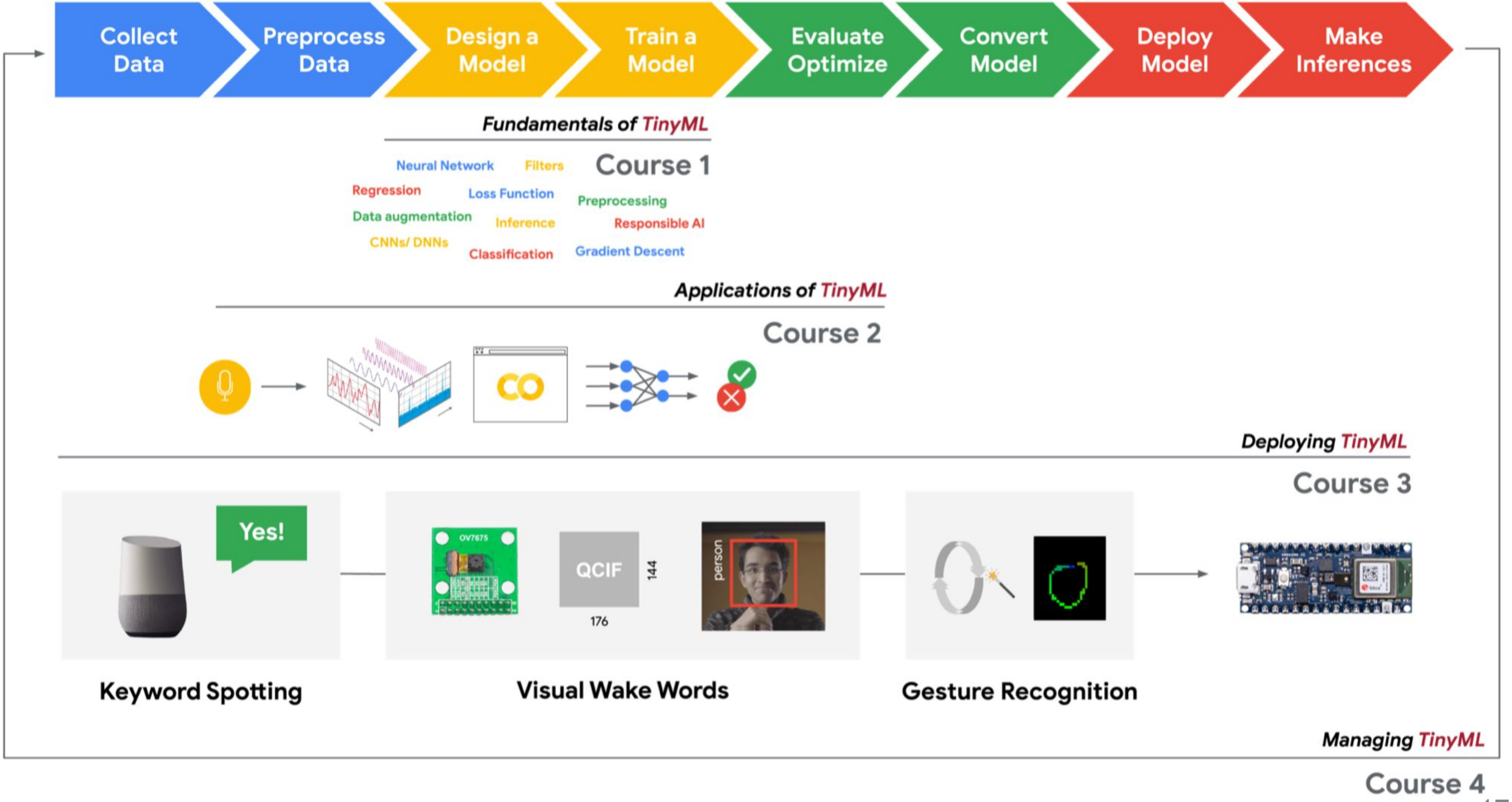


Edge Impulse Simplifies Deployment

The TinyML Workflow



Shows the **score** for **(confidence that the current sounds is)** each of the various keywords and unknown and bolds the highest score.


Collect Data
Preprocess Data
Design a Model
Train a Model
Evaluate Optimize
Convert Model
Deploy Model
Make Inferences
Fundamentals of TinyML
Course 1
Neural Network
Filters
Regression
Loss Function
Preprocessing
Data augmentation
Inference
Responsible AI
CNNs/ DNNs
Classification
Gradient Descent
Applications of TinyML
Course 2
CO
Deploying TinyML
Course 3
Yes!
OV7675
QCIF
144
176
person
Keyword Spotting
Visual Wake Words
Gesture Recognition
Managing TinyML
Course 4

Better Data = Better Models!

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

Hands on Embedded ML
(Vision and Audio)

Brian Plancher
Harvard John A. Paulson School of Engineering and Applied Sciences
brianplancher.com