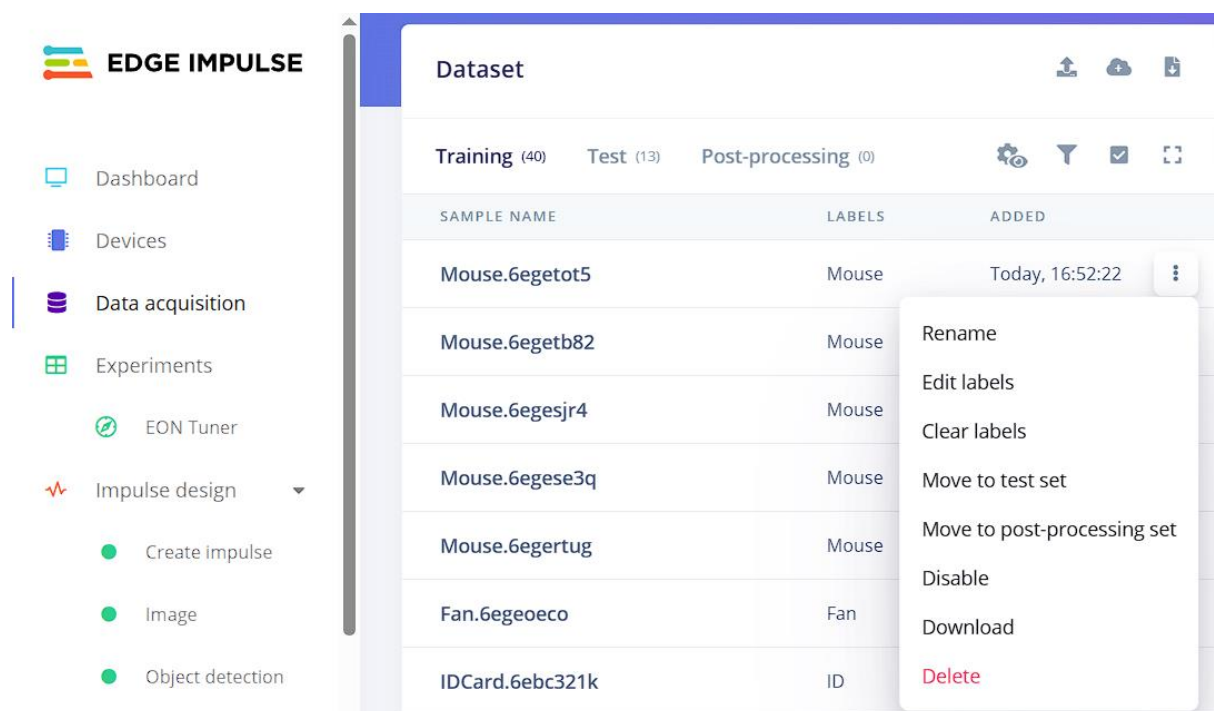


## Edge In-Lab part 2

Adithyash BC

25MML0002

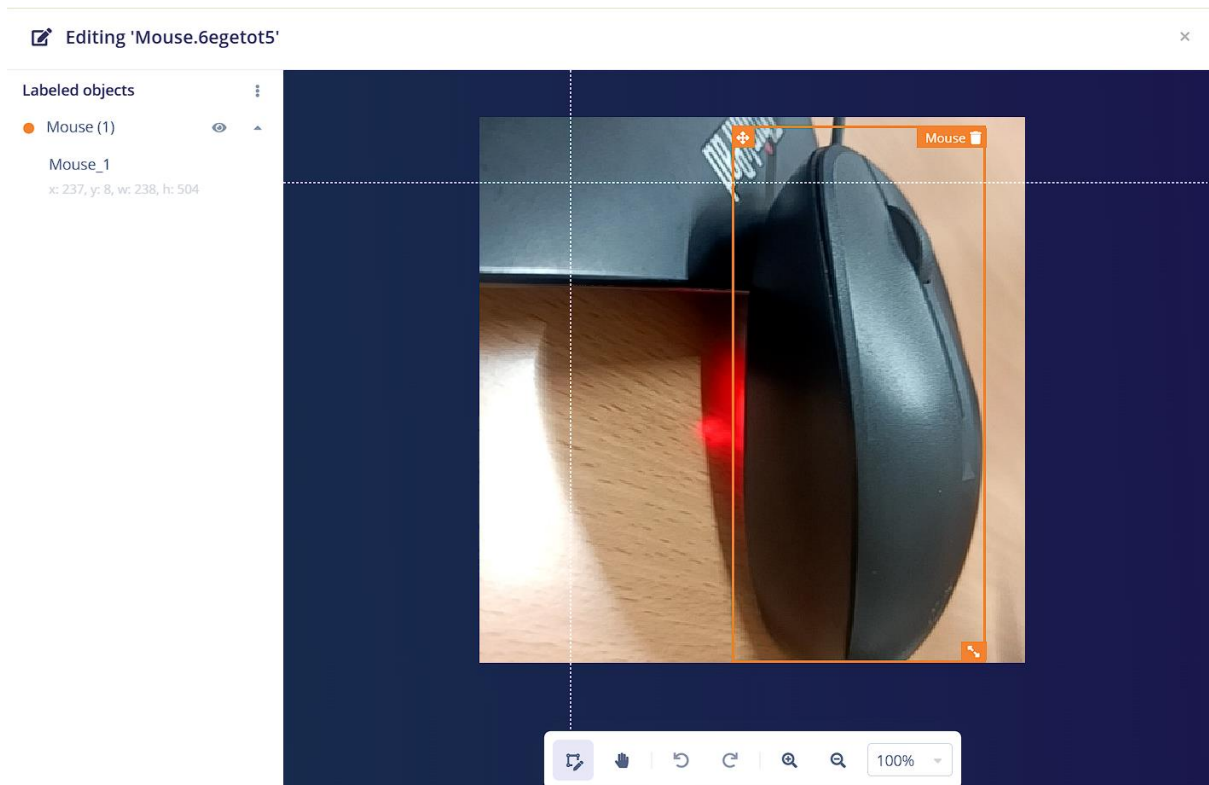
After the data has been acquired, it must be labelled. This can be done by navigating to the 3 dots near the images and selecting it and clicking on edit labels.



The screenshot displays the Edge Impulse web interface. On the left is a sidebar with navigation options: Dashboard, Devices, Data acquisition (selected), Experiments, EON Tuner, and Impulse design. The main area is titled 'Dataset' and shows a table with columns: SAMPLE NAME, LABELS, and ADDED. The table lists several samples, mostly labeled 'Mouse'. A context menu is open for the first sample, 'Mouse.6egetot5', showing options: Rename, Edit labels, Clear labels, Move to test set, Move to post-processing set, Disable, Download, and Delete.

SAMPLE NAME	LABELS	ADDED
Mouse.6egetot5	Mouse	Today, 16:52:22
Mouse.6egetb82	Mouse	
Mouse.6egesjr4	Mouse	
Mouse.6ege3eq	Mouse	
Mouse.6egertug	Mouse	
Fan.6egeoeeco	Fan	
IDCard.6ebc321k	ID	

A window will pop up of the image. You can use the “plus” shaped crosser to box out the object to be labelled. Once boxed a pop up window will show up allowing you to change/make the label. This will help to better inform the model of what it is looking for.



Repeat this process with all the images in the dataset

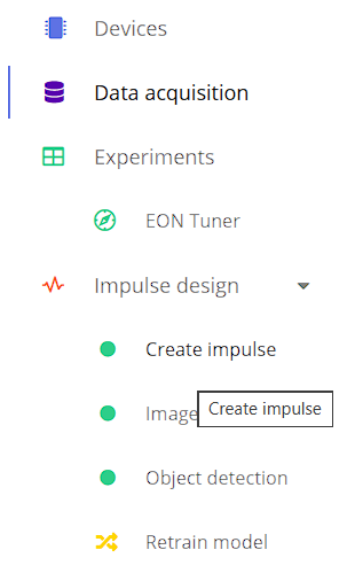
You will know if the label has been placed by looking at the “label” column in the dataset view

LABELS
Mouse
Mouse
Mouse
Mouse
Mouse
Fan
ID
ID
ID
ID
ID

Ensure that the dataset is split into training and testing. The provided infographics on top of the dataset should be able to guide you into better managing this split



Once the data has been optimally split, navigate to the impulse tab and select on create an impulse



Click the “Add a processing block” button to select a format the model can process. Once that is done, click the “Add a learning block” button to select a model



## ⚡ Add a processing block

×

**Did you know?** You can [bring your own DSP code](#).

DESCRIPTION	AUTHOR	RECOMMENDED
<b>Image</b> <small>OFFICIALLY SUPPORTED</small> Preprocess and normalize image data, and optionally reduce the color depth.	Edge Impulse	★ <a href="#">Add</a>
<b>EEG</b> <small>OFFICIALLY SUPPORTED</small> Filters noise and extracts spectral power features from EEG signals.	Edge Impulse	<a href="#">Add</a>

Some processing blocks have been hidden based on the data in your project. [Show all blocks anyway](#)

[Add custom block](#)

[Cancel](#)

## 🧪 Add a learning block

×

**Did you know?** You can [bring your own model](#) in PyTorch, Keras or scikit-learn.

DESCRIPTION	AUTHOR	RECOMMENDED
<b>Object Detection (Images)</b> <small>OFFICIALLY SUPPORTED</small> Fine tune a pre-trained object detection model on your data. Good performance even with relatively small image datasets.	Edge Impulse	<a href="#">Add</a>
<b>Visual Anomaly Detection - FOMO-AD</b> <small>OFFICIALLY SUPPORTED</small> Detect visual anomalies. Extracts visual features using a pre-trained backbone, and applies a scoring function to evaluate how anomalous a sample is by comparing the extracted features to the learned model. Does not require anomalous data.	Edge Impulse	<a href="#">Add</a>

⚡ Want access to all learning blocks? [Upgrade now](#).

Some learning blocks have been hidden based on the data in your project. [Show all blocks anyway](#)

Once that is done, click on “the Save impulse” button to save the setup

The final view should look like this

Impulse #1

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

**Image**

Name  
Image

Input axes (1)  
Image  
image

**Object Detection (Images)**

Name  
Object detection

Input features  
☒ Image

Output features  
5 (Bottle, Fan, ID, Keyboard, Mouse)

**Output features**


5 (Bottle, Fan, ID, Keyboard, Mouse)

Save Impulse

Navigate to the Image tab(below the impulse) and then save the parameters for the raw data. The parameters can be to process the image in RGB or in grayscale. Click on “save parameters” efore you proceed any further.

Raw data

Show: All labels Mouse.6egetot5 (Mous



**Raw features**

0x725931, 0x6f562e, 0x6e552d, 0x624a21, 0x6c522a, 0x6f522b, 0x664722, 0x57328d, ...

**Parameters**


Image

Color depth ②  
RGB

Save parameters

**DSP result**

Image



**Processed features**

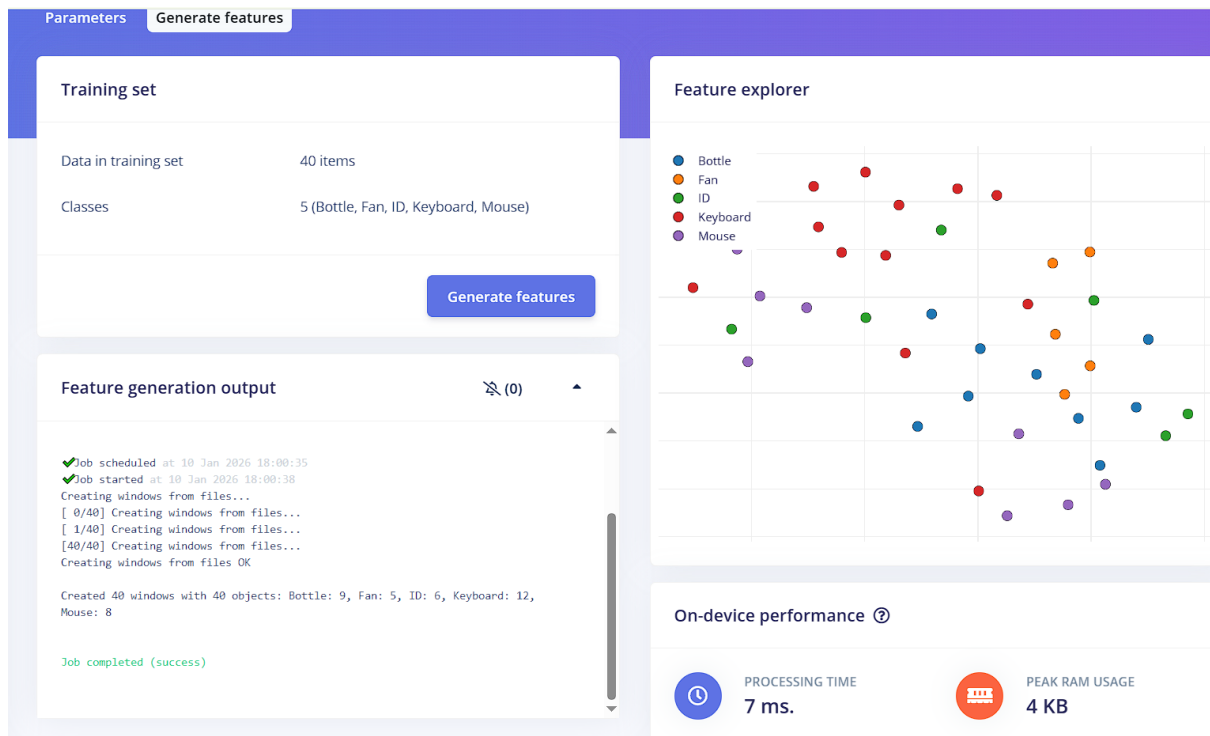
0.4471, 0.3490, 0.1922, 0.4353, 0.3373, 0.1804, 0.4314, 0.3333, 0.1765, 0.3843, ...

**On-device performance**

PROCESSING TIME  
7 ms.

PEAK RAM USAGE  
4 KB

The tab will automatically navigate to the generate features tab. Once there you must click on generate features to load the features of your dataset. After it is done it should look a little something like this



We can then navigate to the final tab, the “Object Detection” tab to run the model on our dataset.

There are various settings that can be found in the neural networks column that will allow you, the end user, to fine tune the model to your liking. Once all the nuances are dialled in( in this case we are just running it plain with no added changes), click on the “Save and train” button by navigating to the bottom of the page.

### Neural Network settings

#### Training settings

Number of training cycles ?

60

Use learned optimizer ?

☐

Learning rate ?

0.001

Training processor ?

CPU


Data augmentation ?

☒

#### Advanced training settings

Neural network architecture

Input layer (27,648 features)



FOMO (Faster Objects, More Objects) MobileNetV2 0.35



Choose a different model

Output layer (5 classes)

Save & train

After a few minutes the Model should spit out the outputs, which can be found on the right-hand side of the page(under training outputs). Take note of the results and make corrections to the model or dataset if need be.

### Training output

 (0) 

```
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
Calculating inferencing time OK
Calculating float32 accuracy...
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
Calculating inferencing time OK
Calculating float32 accuracy...
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
Calculating int8 accuracy...
Extracting TensorBoard logs...
Extracting TensorBoard logs OK

Model training complete

Job completed (success)
```

Model

Model version: ⓘ

Quantized (int8) ▾

Last training performance (validation set)



F1 SCORE ⓘ

0.0%

Confusion matrix (validation set)

	BACKGROU	BOTTLE	FAN	ID	KEYBOARD	MOUSE
BACKGROUN	100%	0%	0%	0%	0%	0%
BOTTLE	100%	0%	0%	0%	0%	0%
FAN	100%	0%	0%	0%	0%	0%
ID	100%	0%	0%	0%	0%	0%
KEYBOARD	100%	0%	0%	0%	0%	0%
MOUSE	100%	0%	0%	0%	0%	0%
F1 SCORE	1.00	0.00	0.00	0.00	0.00	0.00

Metrics (validation set)



METRIC	VALUE
Precision (non-background) ⓘ	0.00
Recall (non-background) ⓘ	0.00
F1 Score (non-background) ⓘ	0.00

In our case the model gave us an F1 score of 0, which means that we will have to tune our dataset and model again to get our desired outputs.