Maple Sung
CART451

Edge Impulse Excercise

1. Initial datasets: Main Objects
    a. Strainer (tea) x 42
    b. Strap with buckle x 55
    c. Other (ex. Shoes, mouse, helmet, key, lights, outlets, floss, etc.) x 80
- Total Object data collected = 177

a.

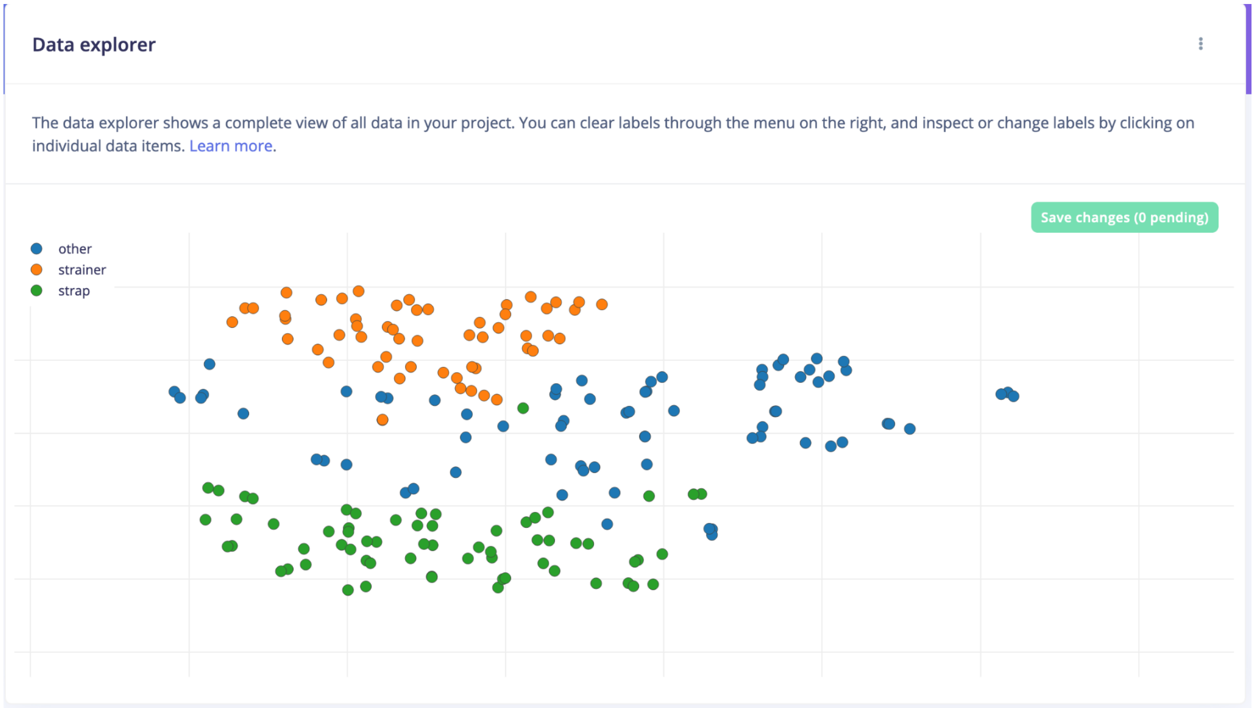
b.

2. This exercise aimed to learn how to upload data onto Edge Impulse. Edge Impulse helps extract a dataset's specific features, classifies them, and gives us visual 2d plots that show me how coherent or poor my data is. Then train a model, see how accurate it is, and learn to optimize the data. Finally, Edge Impulse allows for the gathering of data and deployment of a model in real time.

3. Steps:
- Import dataset (examine data)
- Set type of processing for feature extraction + type of feature for classification
- Set parameters for the type of raw data
- Generate features (examine visual data for discrepancies/weird things)
- Train model with chosen NeuroNetworks and optimized for conditions (speed, accuracy etc)
    - Examine the confusion matrix, if 100% accurate = issues
    - Re-train, replace/remove data if necessary
- Test Model (examine model extractor)
- Deployment: directly on browser/device/library

4. I believe Edge Impulse was able to decipher my dataset was not precise because I had multiple images of the same object after the object was altered. I saw that in Image Generator I had the same data points cluster in 2 different spots. However, was surprised to see that the trained model was still able to recall at a very high accuracy rate abate with very separate features.

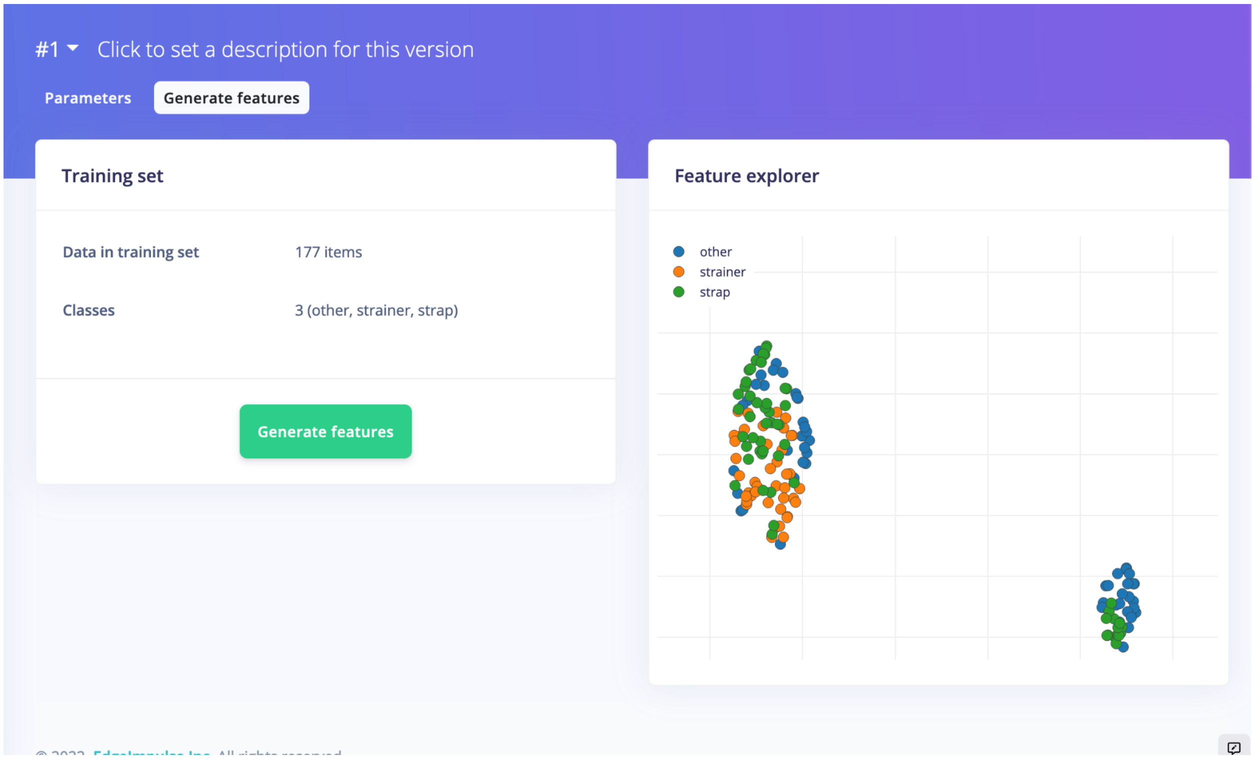
5.



Data explorer

The data explorer shows a complete view of all data in your project. You can clear labels through the menu on the right, and inspect or change labels by clicking on individual data items. Learn more.

Each labeled object is within the same field but not clustered together. This means my data’s features are close but not enough?



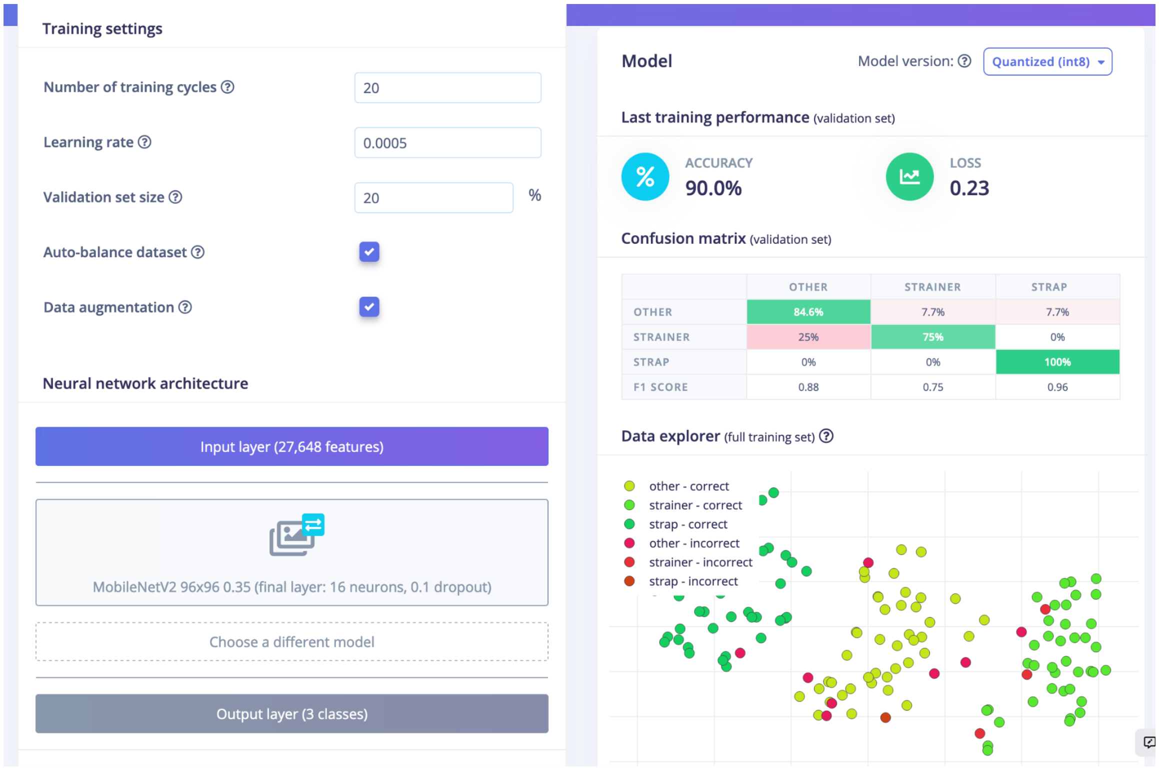
#1 Click to set a description for this version

Training set

Data in training set	177 items
Classes	3 (other, strainer, strap)

Feature explorer

2 different cluster means the same object from the same label but photos taken at the altered position.



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)

Choose a different model

Output layer (3 classes)

Model — Model version: Quantized (int8)

Last training performance (validation set)

ACCURACY 90.0%

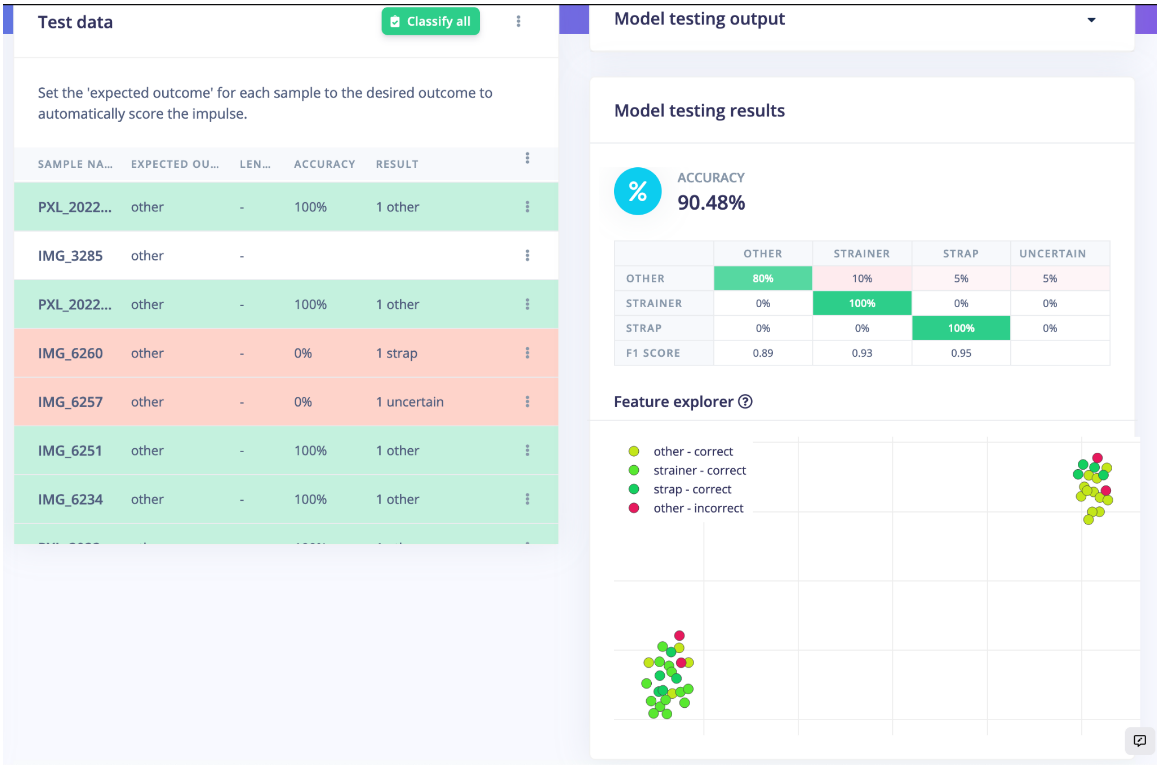
LOSS 0.23

Confusion matrix (validation set)

	OTHER	STRAINER	STRAP
OTHER	84.6%	7.7%	7.7%
STRAINER	25%	75%	0%
STRAP	0%	0%	100%
F1 SCORE	0.88	0.75	0.96

Data explorer (full training set)

The trained model provides very good accuracy but may need to examine data for the strap due to its 100% accuracy.



Test data

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

SAMPLE NA...	EXPECTED OU...	LEN...	ACCURACY	RESULT
PXL_2022...	other	-	100%	1 other
IMG_3285	other	-		
PXL_2022...	other	-	100%	1 other
IMG_6260	other	-	0%	1 strap
IMG_6257	other	-	0%	1 uncertain
IMG_6251	other	-	100%	1 other
IMG_6234	other	-	100%	1 other

Model testing output

Model testing results

ACCURACY 90.48%

	OTHER	STRAINER	STRAP	UNCERTAIN
OTHER	80%	10%	5%	5%
STRAINER	0%	100%	0%	0%
STRAP	0%	0%	100%	0%
F1 SCORE	0.89	0.93	0.95	

Feature explorer

When the model was tested, the strainer and the strap has an accuracy rate of 100%. This leads me to think that my initial dataset for training the model is too small and doesn't contain enough samples. Or due to the poor data samples, the algorithm learns the noise of the data instead what I want it to recognize.

6. I could first using the trained model data explorer, remove the data the model does not recognize as any of the labeled objects. Or I could re-do another project with better data samples to see how it would affect the trained model and testing result.

PART C

I would use Object Detection to determine (with a percentage) what plants and fungi are safe to eat if I ever need to live off the land. With unlimited battery life (obviously), every time I scan a forest floor, anything edible would show up.

Ideally, I would have a model trained for different continents and climates. My data would be taken from all known public visual resources of edible plants and mushrooms. My labels would be 'leafy', 'roots' (carbs), and 'fungai'. At this point, I'd hope that I would have enough data samples so my models give me high accuracy for detection.