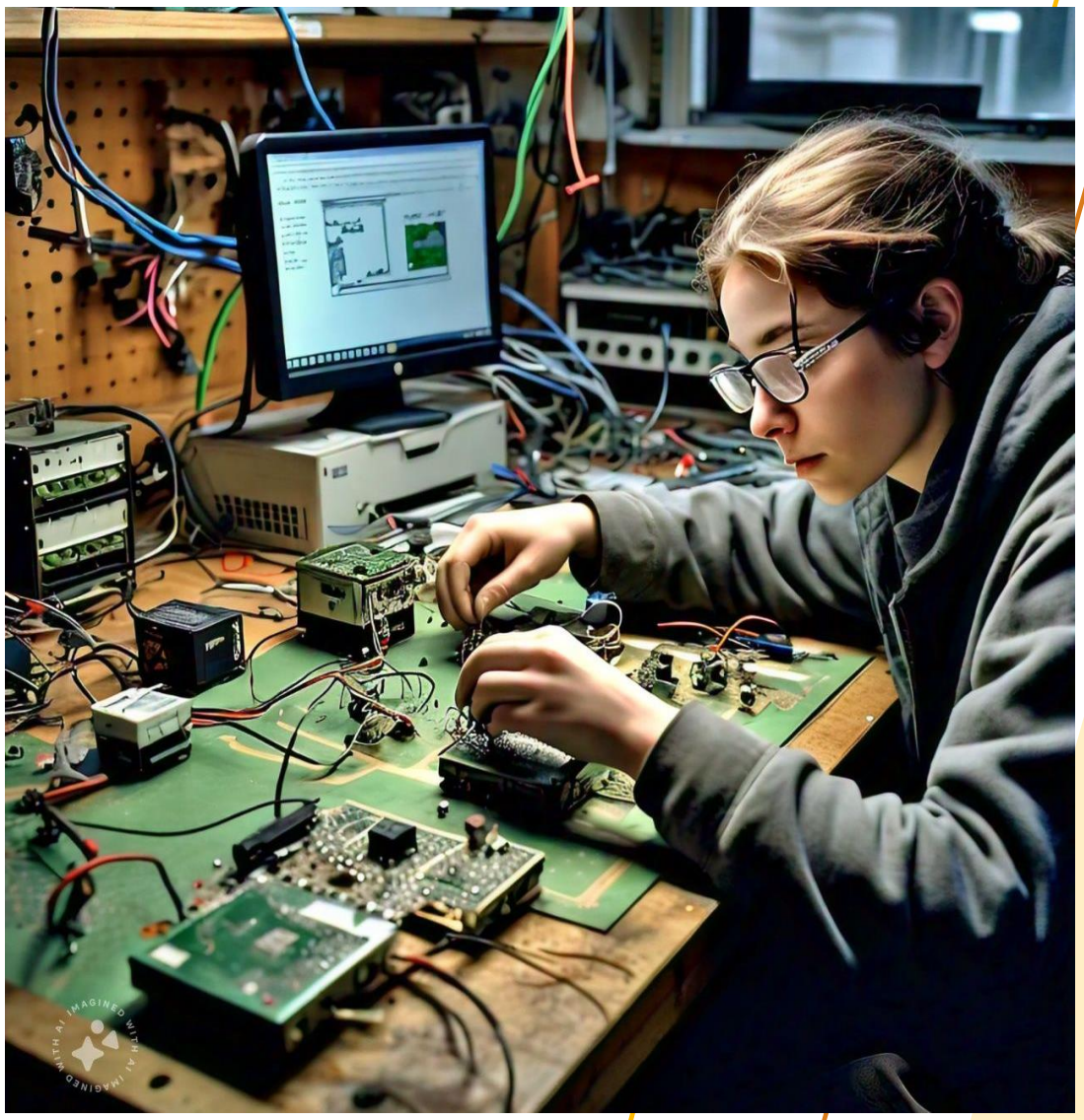


AutoBill

An innovative AI-powered autonomous checkout system designed to revolutionize the retail experience. AutoBill harnesses the power of cutting-edge technology to streamline the checkout process, eliminating long queues and enhancing customer satisfaction.

Steps

01. Connections

02. Edge Impulse

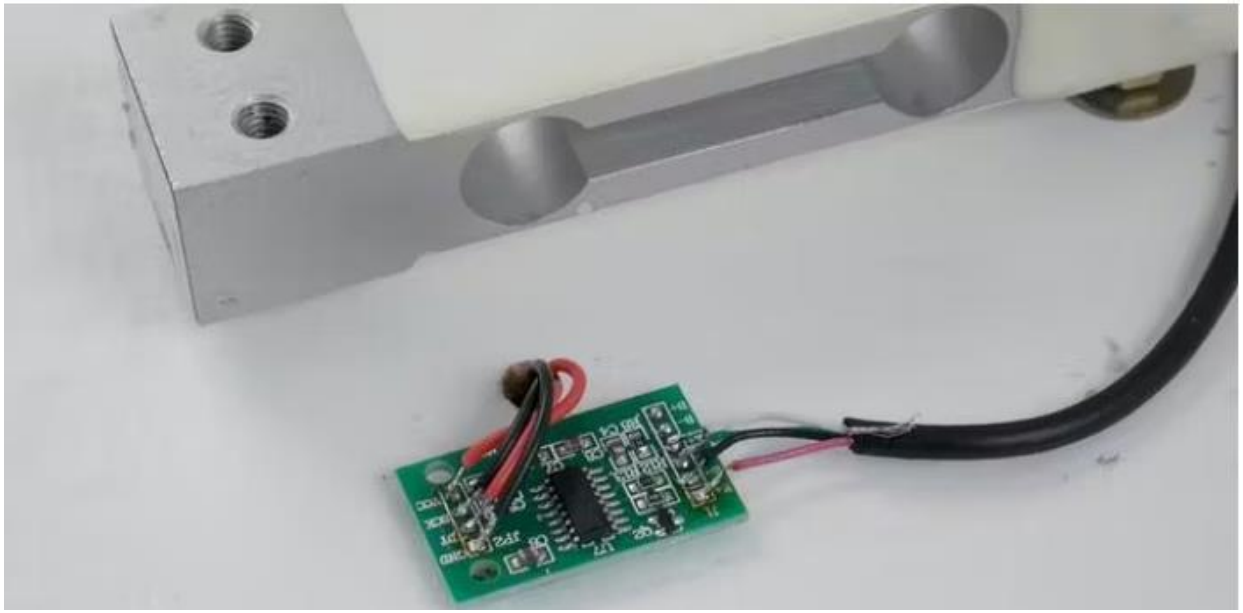
03. Developing the Checkout interface

Step One

Connections

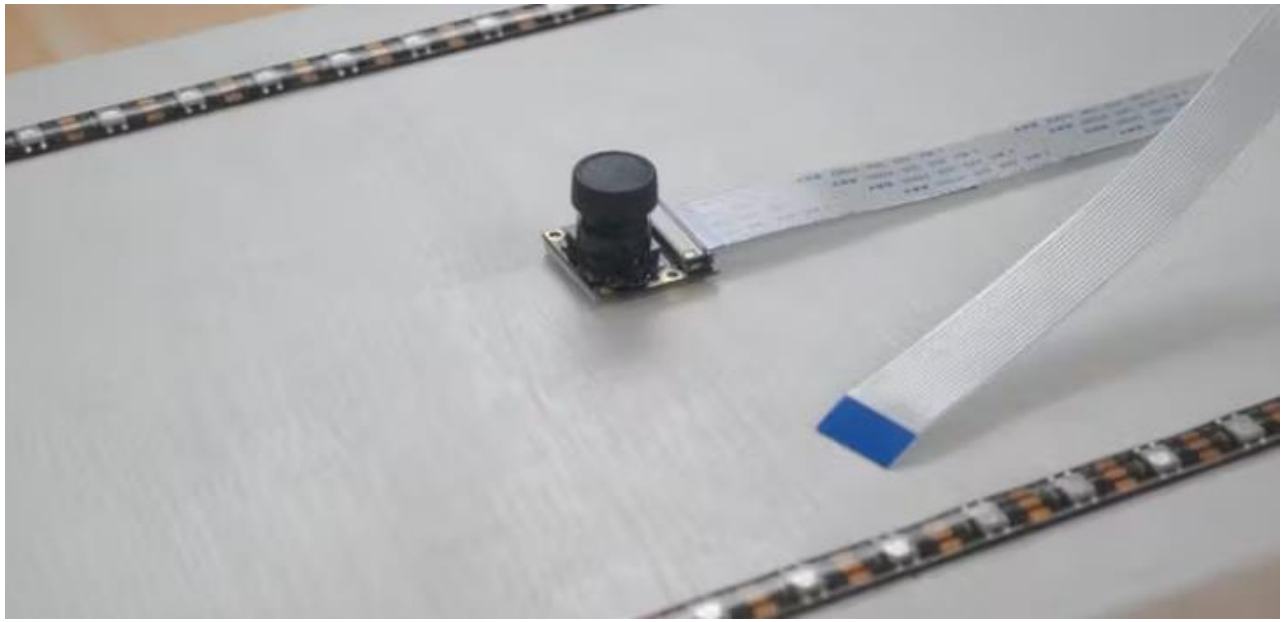
- Cabinetry
- load cell to the HX711
- Camara module to cabinent hole and led strip
- Raspberry pi connections

Load cell and amplifier



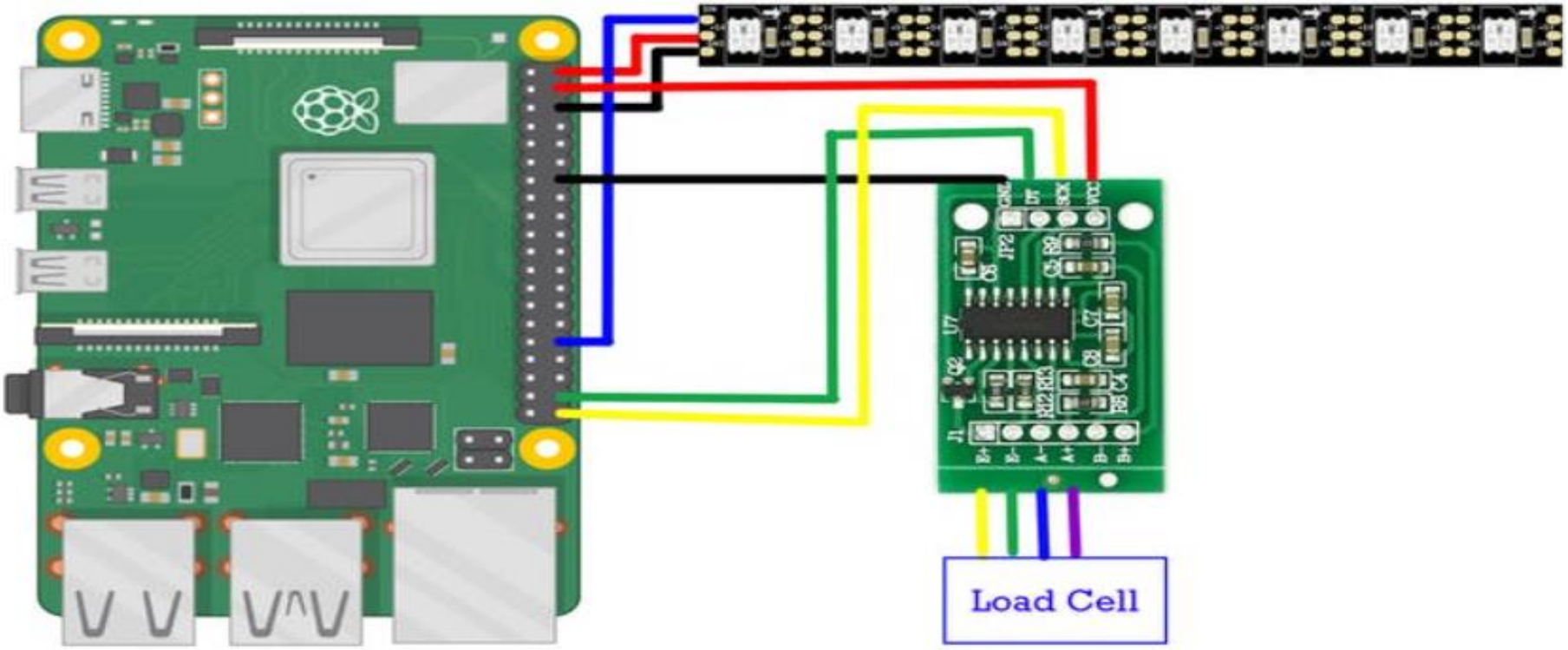
One end of the load cell should be rigidly connected and another end should be floated in the air, then only we get the proper weight of the object. Then we wired the load cell to the HX711 break-out module by installing it.

Camera Installation

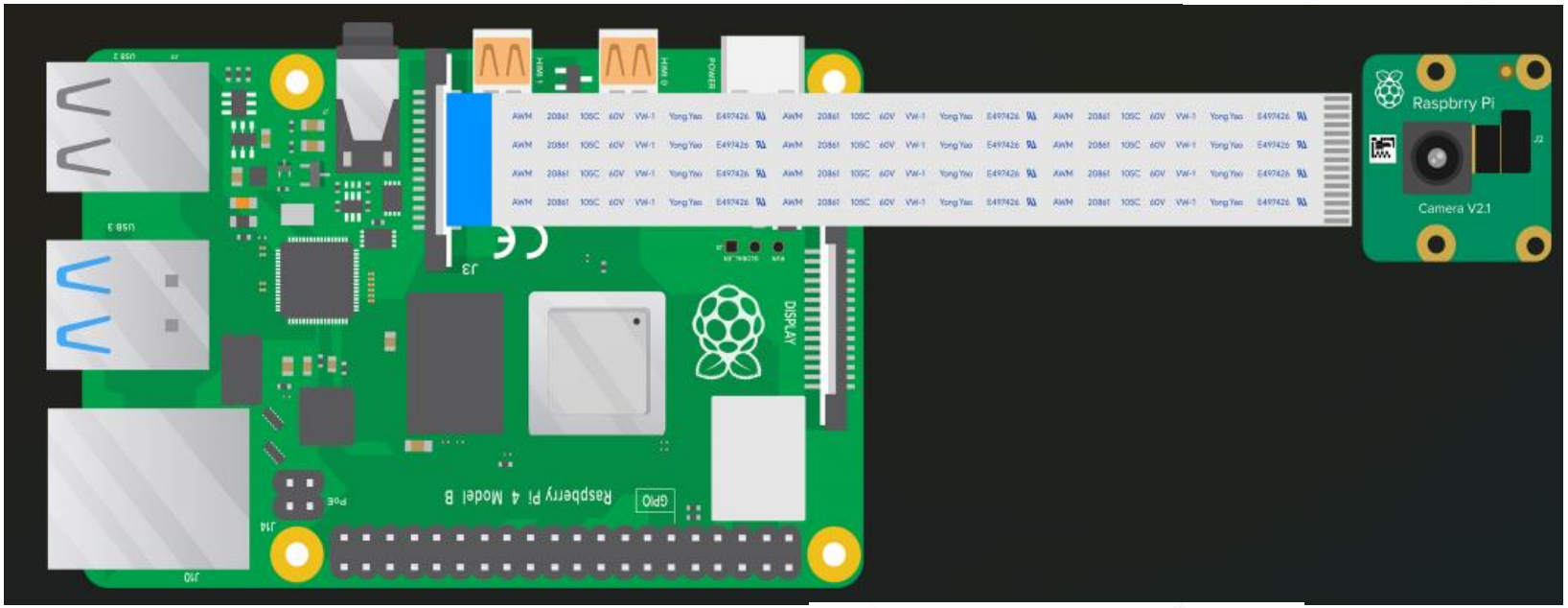


we installed the camera module at the opposite face of the load cell.Then we attached the led strip parallel to the camera module.

Raspberry Pi with load cell and led strip



Raspberry pi camera module



Activation of Camera module in raspberry pi

Step two: Edge Impulse

Edge Impulse is one of the leading development platforms for machine learning Here we are using machine learning to build a system that can recognize the products available in the shops. While training the model, the product images are used to let the model distinguish between them. The images are captured and used in datasets.

Edge Impulse is a powerful platform that enables developers to build intelligent embedded systems with ease.

1 **Impulse creation**

Involves Collecting Sensor data

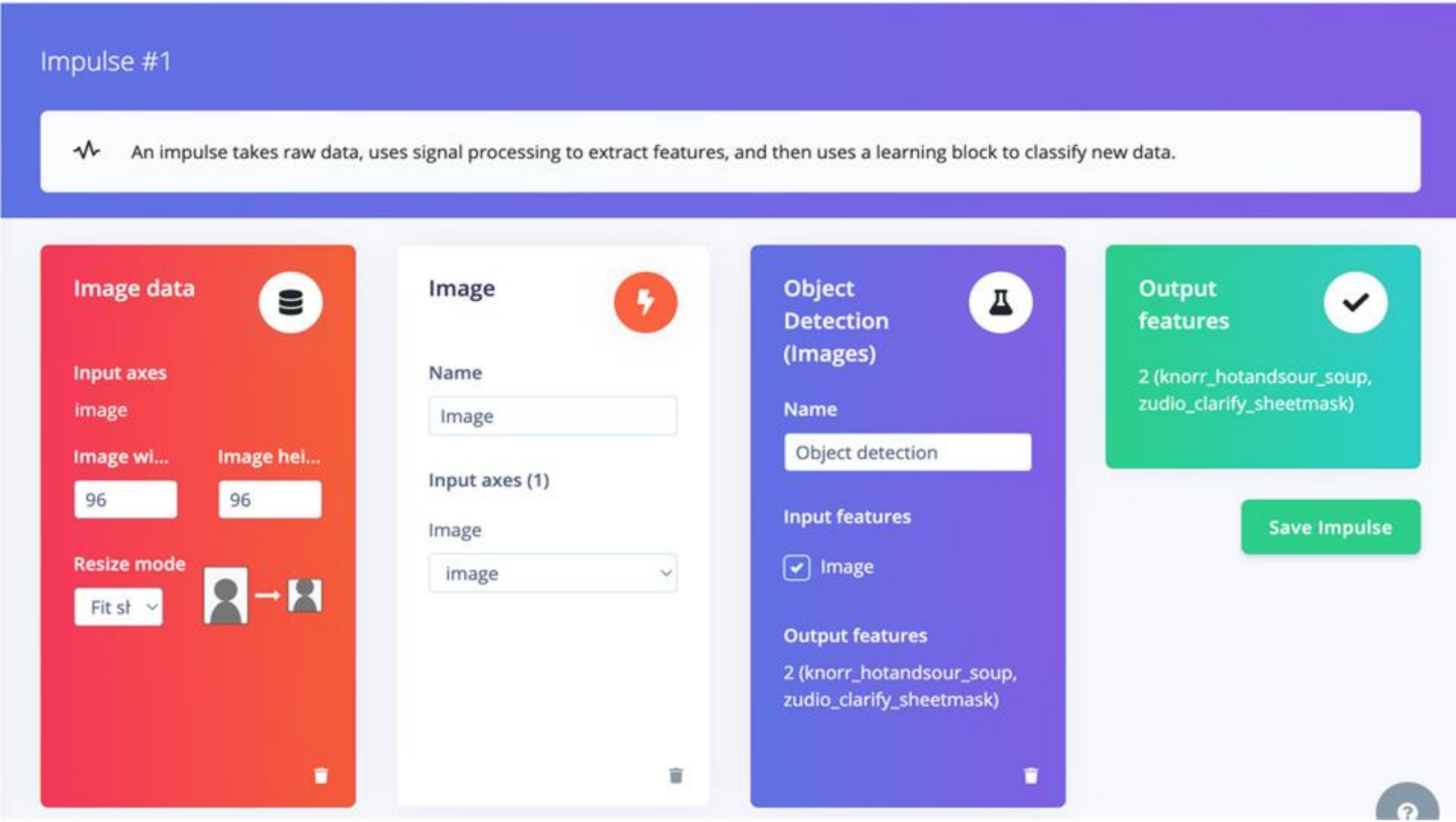
2 **Object Detection**

Facilitates the identification and localization of multiple objects within an image

3 **Live classification**

Enabiling real time data analysis by applying trained models

EDGE IMPULSE


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 wi...
Image hei...
96
96
Resize mode
Fit sl
Image
Name
Image
Input axes (1)
Image
image
Object Detection (Images)
Name
Object detection
Input features
Image
Output features
2 (knorr_hotandsour_soup, zudio_clarify_sheetmask)
Output features
2 (knorr_hotandsour_soup, zudio_clarify_sheetmask)
Save Impulse

The image shows a visual representation of an "Impulse" within a machine learning workflow. It breaks down the process into four distinct steps: Image data, Image, Object Detection (Images), and Output Features.

Image data represents the input data used for training the model. It includes information like the image dimensions.

Image refers to the actual image being processed.

Object Detection focuses on identifying specific objects within the image.

Output Features are the results of the object detection process, which can be further used for classification or other tasks.

Input axes: This section specifies the dimensions of the image data that will be used for training. The user has set the width and height to 96 pixels. This suggests the model will be trained on a specific image size.

Image: This section refers to the actual image files used for training. The user can either upload images directly or use a dataset that contains images. The process of acquiring these images is data collection.

Resize mode : This option controls how the input images are resized to match the desired size (96x96). Different resize methods can impact the model's performance.

Output features: This section defines the categories the model will learn to identify. The user has defined two categories: "knorr_hotandsour_soup" and "zudio_clarify_sheetmask."

Object Detection (Images): This indicates the model will be trained to identify specific objects within images, rather than just classifying the whole image.

Image: The user has selected the image feature as input. This means the model will use the actual image data to identify objects.The user can either upload images directly or use a dataset that contains images. The process of acquiring these images is data collection.

Parameters | Generate features

Raw data

Show: All labels | WhatsApp Image 2024-10-02 at 15.07.28 (37) (zud

Raw features

0x965a35, 0x995d38, 0x9d5f3a, 0x935934, 0x925833, 0x915531, 0x935733, 0x9c5f3b, 0x985…

Parameters

Image

Color depth ⓘ RGB

Save parameters

DSP result

Image

Processed features

0.5882, 0.3529, 0.2078, 0.6000, 0.3647, 0.2196, 0.6157, 0.3725, 0.2275, 0.5765, 0.349…

On-device performance ⓘ

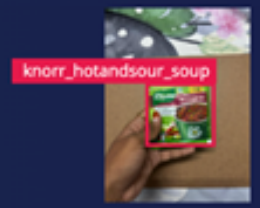
PROCESSING TIME
7 ms.

PEAK RAM USAGE
4 KB

Parameters | Generate features

Raw data

Show: knorr_hotandsour_soup | WhatsApp Image 2024-10-02 at 14.25.36 (50) (kno



Raw features

0x6b7080, 0x646c7b, 0x6b7181, 0x6c7584, 0x747d8c, 0x6d7687, 0x696f82, 0x666b7e, 0x717…

Parameters

Image

Color depth ⓘ RGB

Save parameters

DSP result

Image

Processed features

0.4196, 0.4392, 0.5020, 0.3922, 0.4235, 0.4824, 0.4196, 0.4431, 0.5059, 0.4235, 0.458…

On-device performance ⓘ

PROCESSING TIME
7 ms.

PEAK RAM USAGE
4 KB

Parametres

Generate Features: This is the main functionality of the Edge Impulse platform. It allows users to generate features from their data, which can be used to train machine learning models.

Show: All labels: This dropdown menu allows users to filter the data by specific labels. In this case, the user has selected "All labels" to view all the data

Raw Features: This section displays the raw data that has been uploaded to the platform. The data is represented as a table with various columns, including "0x965a35", "0x995d38", "0x9d5f3a", and so on. These columns likely represent different features or variables in the data.

Image: This section displays an image that is associated with the raw data. The image appears to be a small, square image with a green border.

On-Device Performance:

This section displays the performance metrics of the model running on a device.

Processing Time: 7 ms (milliseconds) - This represents the time taken by the model to process the image. A lower processing time indicates better efficiency.

Peak RAM Usage: 4 KB (kilobytes) - This is the maximum amount of memory used by the model during processing. Lower RAM usage is desirable, as it indicates less strain on the device's resources.

Here Same goes for the other image which is a product label for a "Hot and sour" soup packet by the brand "knorr."

DSP Result:

The DSP result displays the image processed by the model. Here, the image is a product label for a "Salicylic Acid" sheet mask by the brand "Zudio."

DSP (Digital Signal Processing): A technique used to manipulate and analyze signals, particularly in the digital domain. In this context, the DSP likely involves:

Image preprocessing: Enhancing the image for better analysis by the model.

Feature extraction: Identifying important features within the image, like edges, colors, and patterns.

Classification: Analyzing extracted features to determine what the image represents.

Color Depth:

Color depth refers to the number of bits used to represent each color component (red, green, blue) of a pixel. The "RGB" setting indicates the image is using the standard RGB color model.

In simpler terms: This means the image can display a wide range of colors.

Neural Network settings

Training settings

Number of training cycles: 100

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 (2 classes)

Training...

Training output

Cancel (0)

```
Creating job... OK (ID: 24726453)

Scheduling job in cluster...
Container image pulled!
Job started
Scheduling job in cluster...
Container image pulled!
Job started
Splitting data into training and validation sets...
Splitting data into training and validation sets OK

Training model...
Training on 69 inputs, validating on 18 inputs
```

Training output

Cancel (0)

```
Training on 69 inputs, validating on 18 inputs

Epoch   Train    Validation
Loss     Loss    Precision Recall F1
00   1.22274  0.81611 0.00      0.00   0.00

Epoch   Train    Validation
Loss     Loss    Precision Recall F1
01   0.70310  0.54654 0.00      0.00   0.00

Epoch   Train    Validation
Loss     Loss    Precision Recall F1
02   0.46419  0.33307 1.00      0.10   0.17
```

Epoch: An epoch represents one complete pass through the entire training dataset. In this case, the model has been trained for 3 epochs ..

Train Loss: This measures how well the model is performing on the training data during each epoch. A lower train loss generally indicates better performance.

Validation Loss: This is similar to train loss but is measured on a separate dataset called the "validation set". This set is not used during training but allows us to assess how well the model generalizes to unseen data.

Validation Precision, Recall, F1: These are metrics commonly used in machine learning, especially in classification problems, to evaluate the model's performance. Let's break down each:

Precision: This tells us the percentage of instances the model correctly classified as positive out of all instances it predicted as positive.

Recall: This tells us the percentage of actual positive instances the model correctly classified as positive.

F1 Score: This is a combined measure of precision and recall, providing a more holistic view of the model's performance. It's often used when precision and recall need to be balanced.

The provided output suggests:

The model's train loss decreased significantly over the epochs, indicating improvement in learning.

The validation loss also generally went down, suggesting that the model is learning well and not overfitting to the training data.

The precision, recall, and F1 scores show that the model is still underperforming in classification tasks.

Model

Model version: Quantized (int8)

Last training performance (validation set)

F1 SCORE
95.2%

Confusion matrix (validation set)

	BACKGROUND	KNORR_HOTANDSOUR_SC	ZUDIO_CLARIFY_SHEETMA
BACKGROUND	100.0%	0.0%	0%
KNORR_HOTANDSOUR_S	0%	100%	0%
ZUDIO_CLARIFY_SHEETM	12.5%	0%	87.5%
F1 SCORE	1.00	0.96	0.93

Metrics (validation set)

METRIC	VALUE
Precision (non-background)	0.95
Recall (non-background)	0.95
F1 Score (non-background)	

Recall (non-background)

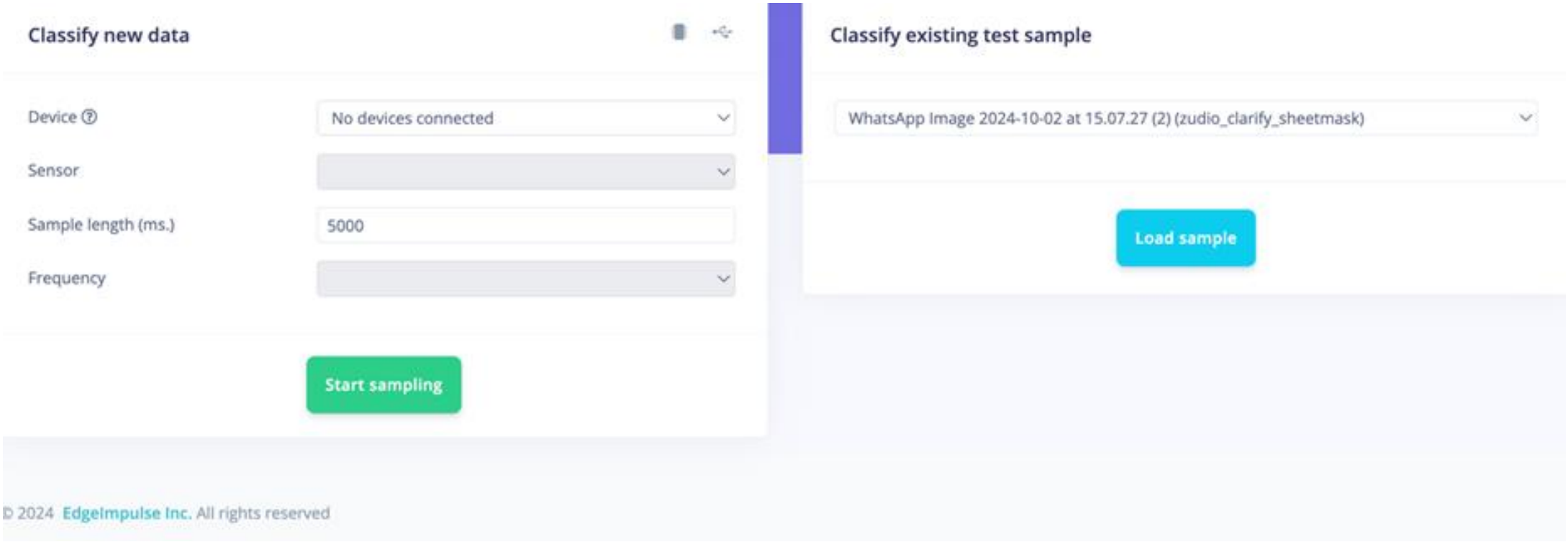
On-device performance

Engine: EON™ Compiler

INFERENCING TIME
707 ms.

PEAK RAM USAGE
239.5K

FLASH USAGE
79.0K



Classification result

Summary

Model version: Unoptimized (float32)

Name: WhatsApp Image 2024-10-02 at 15.07.27 (2)

CATEGORY	COUNT	INFO
F1 score	100.00%	Combines the precision and recall
Precision	100.00%	Predictions made that are correct
Recall	100.00%	Labeled objects that were detected
zudio_clarify_sheetmask	1	

In FOMO, labels are collapsed to centriods, and two close objects may be collapsed to one. ...
View more

RAW DATA / CLASSIFICATION RESULT

WhatsApp Image 2024-10-0...

Overlay

Raw features

```
0x9ba7c5, 0x9da8c6, 0x99a4c2, 0x95a1ba, 0x919bb5, 0x8992ad, 0x8890a8, 0x8d94ab, 0x8c9…
```

View

Ground truth: Show all (1)

Predictions: Show all (1)

Show labels

Summary

Model version: Unoptimized (float32)

Name: WhatsApp Image 2024-10-02 at 14.25.36 (12)

CATEGORY	COUNT	INFO
F1 score	100.00%	Combines the precision and recall
Precision	100.00%	Predictions made that are correct
Recall	100.00%	Labeled objects that were detected
knorr_hotandsour_soup	3	

In FOMO, labels are collapsed to centriods, and two close objects may be collapsed to one. ...
View more

RAW DATA / CLASSIFICATION RESULT

WhatsApp Image 2024-10-0...

Side by side

Raw features

0x71442b, 0x73482f, 0x6d442c, 0x70462e, 0x6e432b, 0x72462d, 0x764a31, 0x764930, 0x764…

View

Ground truth: Show all (3)

Predictions: Show all (3)

Show labels

Training Settings:

Number of Training Cycles: The number of times the network will go through the entire training dataset. This is set to 100.

Use Learned Optimizer: A checkmark indicates that a pre-trained optimizer is being used. This helps the network converge faster.

Learning Rate: Controls how quickly the network adjusts its weights during training. A lower rate is more precise but can be slower. The value is 0.001.

Training Processor: The chosen processor for training is CPU, suggesting this is a smaller training task.

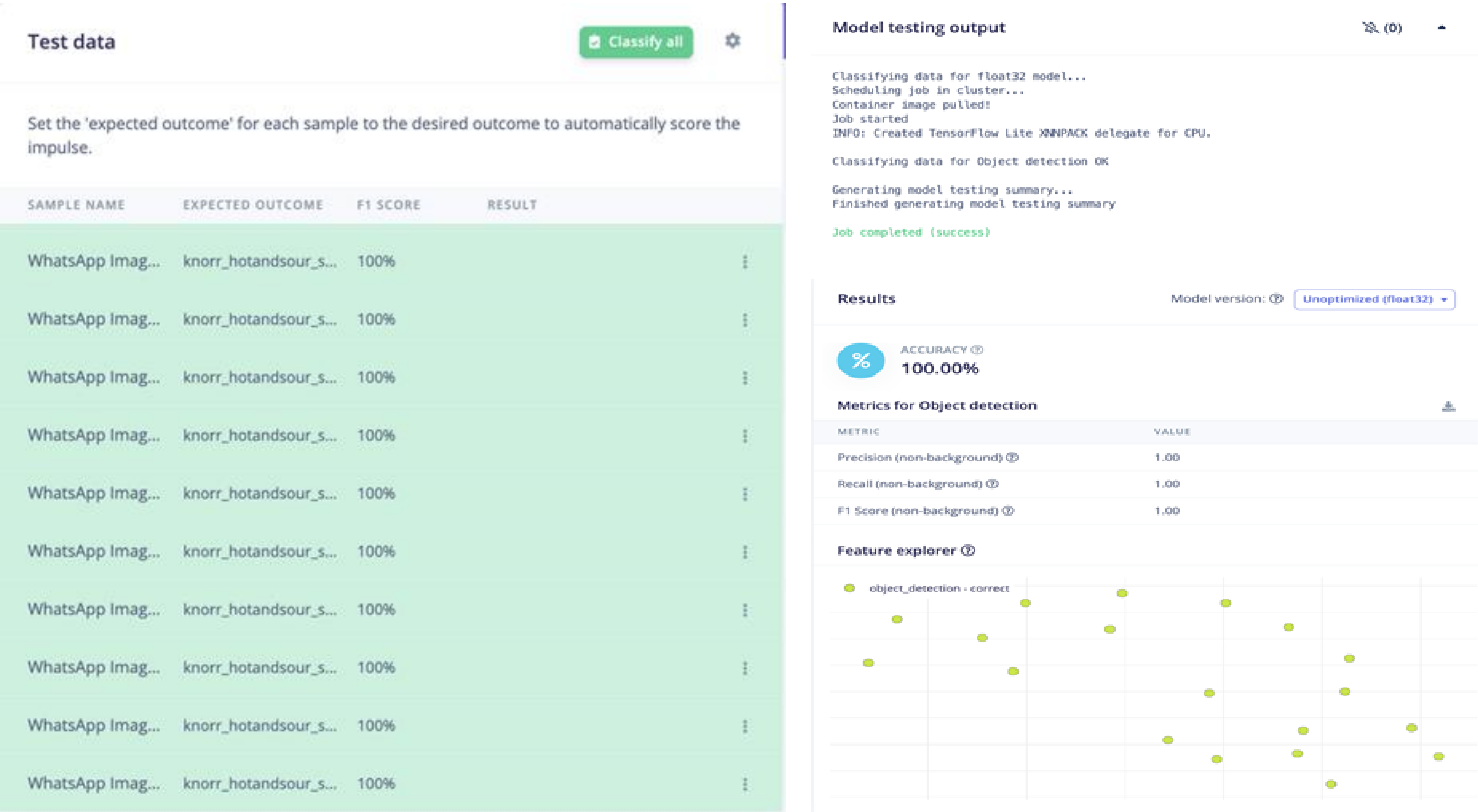
Data Augmentation: Enabled, likely increasing the dataset size by generating variations of existing images (e.g., rotations, flips) to improve the model's robustness.

Neural Network Architecture:

Input Layer: The network takes images as input, and the size is 27,648 features, indicating a substantial amount of image data.

Model Choice: The selected model is FOMO (Faster Objects, More Objects) MobileNetV2 0.35. This suggests the goal is object detection, with a focus on speed (due to MobileNetV2).

Output Layer: The network predicts 2 classes, suggesting a binary classification problem (e.g., "object present" or "object not present").


Test data
Classify all
Set the 'expected outcome' for each sample to the desired outcome to automatically score the impulse.
SAMPLE NAME
EXPECTED OUTCOME
F1 SCORE
RESULT
WhatsApp Imag...
knorr_hotandsour_s...
100%
WhatsApp Imag...
knorr_hotandsour_s...
100%
WhatsApp Imag...
knorr_hotandsour_s...
100%
WhatsApp Imag...
knorr_hotandsour_s...
100%
WhatsApp Imag...
knorr_hotandsour_s...
100%
WhatsApp Imag...
knorr_hotandsour_s...
100%
WhatsApp Imag...
knorr_hotandsour_s...
100%
WhatsApp Imag...
knorr_hotandsour_s...
100%
WhatsApp Imag...
knorr_hotandsour_s...
100%
WhatsApp Imag...
knorr_hotandsour_s...
100%
Model testing output
(0)
Classifying data for float32 model...
Scheduling job in cluster...
Container image pulled!
Job started
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
Classifying data for Object detection OK
Generating model testing summary...
Finished generating model testing summary
Job completed (success)
Results
Model version:
Unoptimized (float32)
ACCURACY
100.00%
Metrics for Object detection
METRIC
VALUE
Precision (non-background)
1.00
Recall (non-background)
1.00
F1 Score (non-background)
1.00
Feature explorer
object_detection - correct

Metric

This column lists the different metrics used to evaluate the performance of the model. These metrics are:

Background: This refers to the background of the image, which is not an object of interest.

Zudio Clarify Sheetmask: This is a specific type of object that the model is trying to detect.

Knorr Hot and Sour Soup: This is another specific type of object that the model is trying to detect.

Value

This column shows the actual values for each metric. The values are:

0.95: This is the value for the background metric, indicating that the model is very accurate in detecting the background.

0.93: This is the value for the Zudio Clarify Sheetmask metric, indicating that the model is very accurate in detecting this type of object.

0.96: This is the value for the Knorr Hot and Sour Soup metric, indicating that the model is very accurate in detecting this type of object.

Precision (non-background)

This column shows the precision of the model for each metric, excluding the background. Precision is a measure of how accurate the model is in predicting the correct class. The values are:

0.95: This is the precision for the Zudio Clarify Sheetmask metric, indicating that the model is very accurate in predicting this type of object.

0.96: This is the precision for the Knorr Hot and Sour Soup metric, indicating that the model is very accurate in predicting this type of object.

Recall (non-background)

This column shows the recall of the model for each metric, excluding the background. Recall is a measure of how well the model detects all instances of a particular class. The values are:

0.95: This is the recall for the Zudio Clarify Sheetmask metric, indicating that the model is very good at detecting all instances of this type of object.

0.96: This is the recall for the Knorr Hot and Sour Soup metric, indicating that the model is very good at detecting all instances of this type of object.

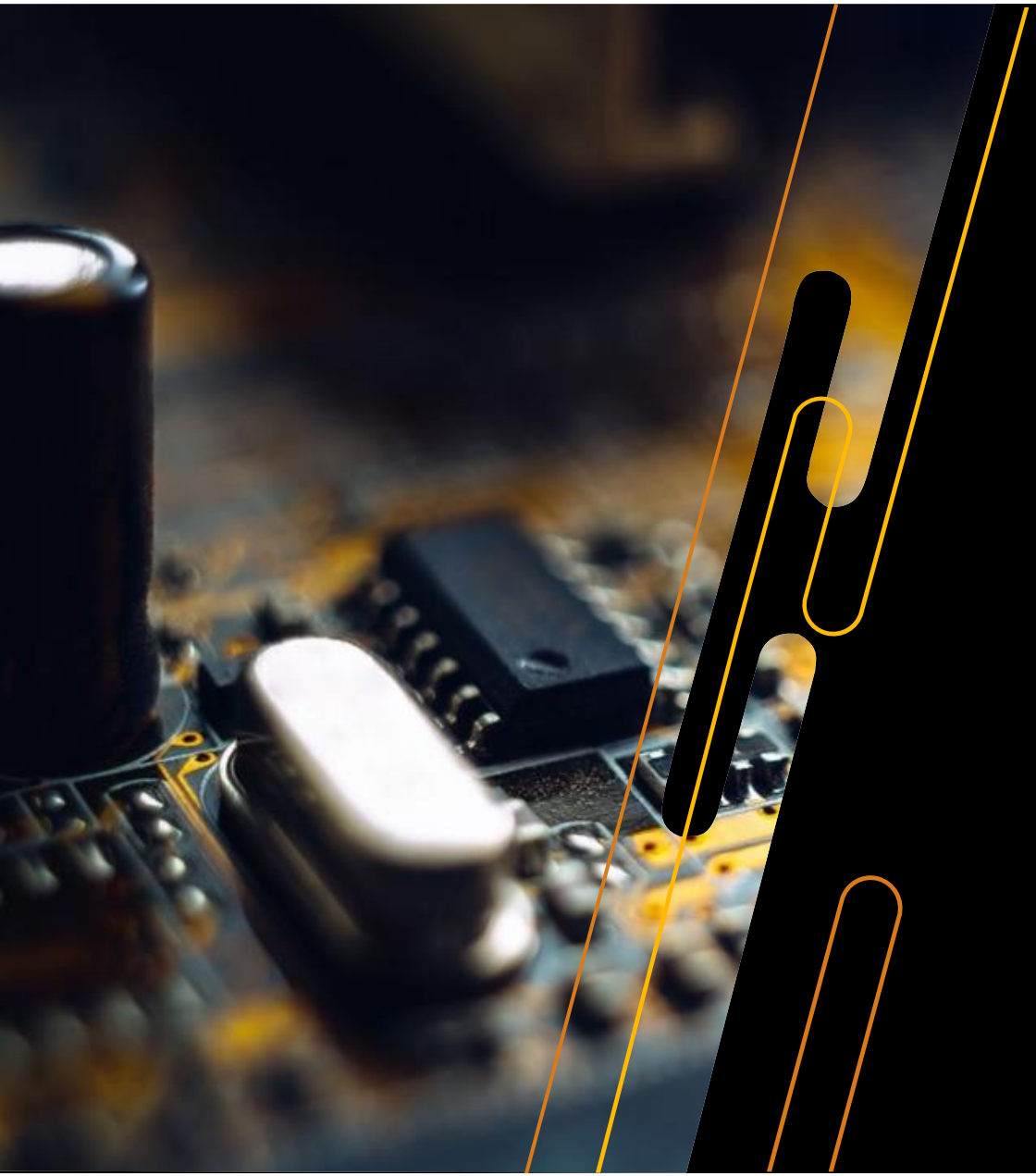
Overall, the table suggests that the model is very accurate in detecting the background and the two specific types of objects (Zudio Clarify Sheetmask and Knorr Hot and Sour Soup). The precision and recall values are all very high, indicating that the model is performing well in terms of accuracy and detection.

Step Three: Developing the checkout Interface

Checkout Interface

The checkout interface has two parts,

1. Front-end developed using HTML, JS
2. Backend API developed using NodeJS and Express



Credits

23BCE8934 – PPT , Object Detection
23BCE8939 - Front end
23BCE7913 – PPT , Classification
23BCE9728 – Impulse Creation
23MIC7234 - Connections
20MIS7041 – Front end


Thank you