Hochschule Emden/Leer
Department of Electrical Engineering and Computer Science
Industrial Informatics

University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Machine Vision with Arduino Portenta H7

Face recognition using Edge Impulse and Arduino PortentaH7 connected to Vision Shield.

Vatsal Mahajan, Manoj Selvaraju, Vijay Singh

September 11, 2024

Table of Content I

Introduction

Edge Impulse offers a cloud-based system for data collection, neural network design, model training, testing and deployment.

The key steps are:

- Dataset Preparation and Upload.
- Choose the best model for the task.
- Model Training.
- Model Testing and Export.

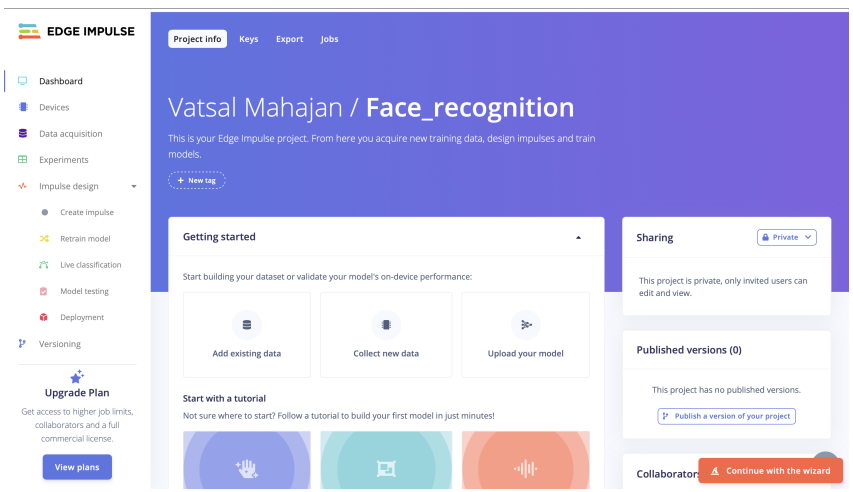


Figure 1: Edge Impulse

Dataset Preparation and Upload

Creating Your Own Dataset

In the Project we are using our won dataset for the model training and testing.

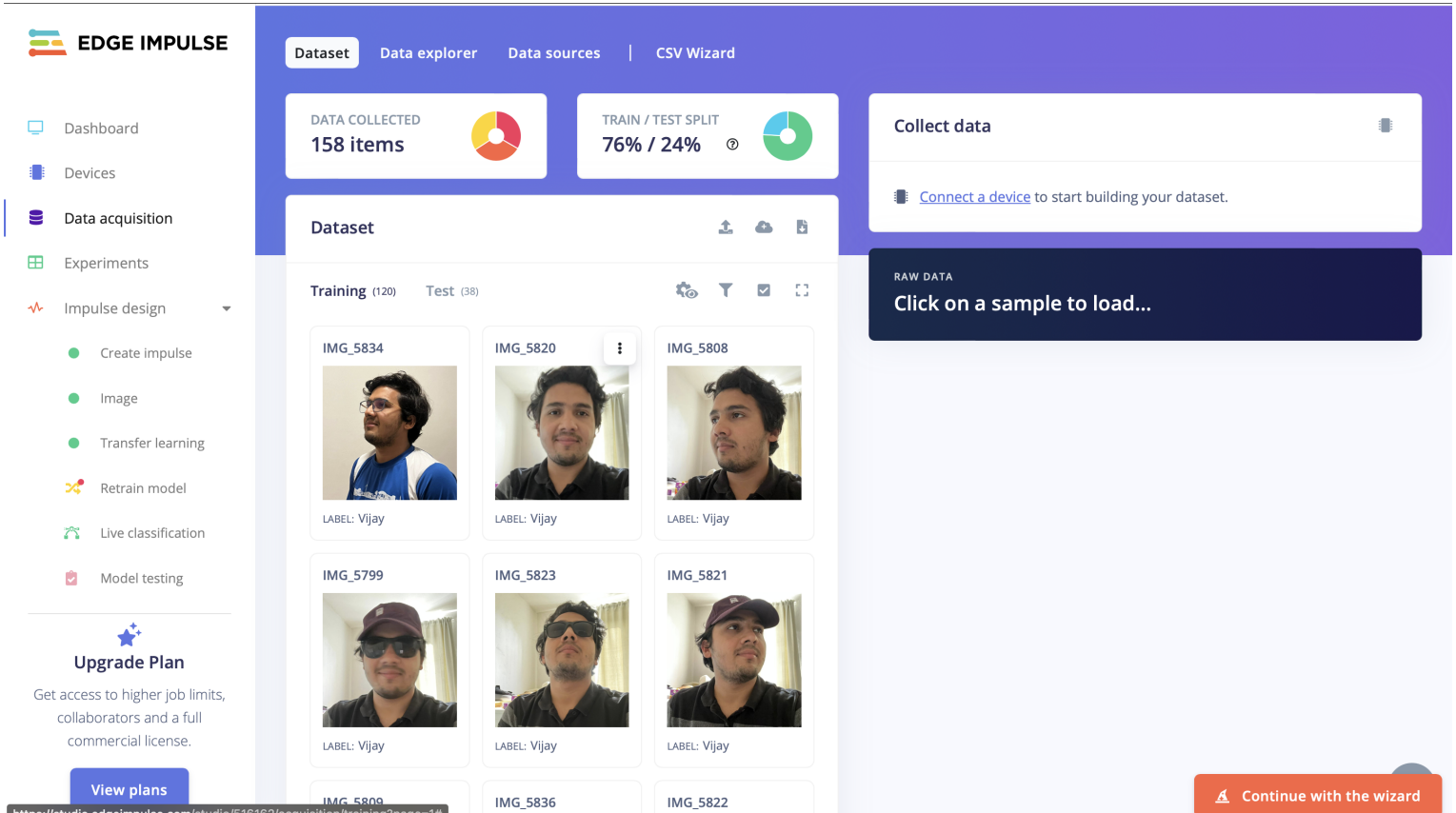
- We have face images for the three distinct people, organize them in separate folders named Vatsal, Manoj, and Vijay.
- To train the model we need the enough images (at least 50-70 per person) under different conditions (lighting, angle) for better performance.

Uploading the dataset to Edge Impulse

- Go to Edge Impulse and log in or create an account.
- Create a new project, name it (e.g.,"Face Recognition"), and select Image as the type of data for your project.
- Go to the "Data Acquisition" tab: This is where you'll upload your dataset.

Uploading Images

- Click on Upload Data. You can drag and drop the images of each person into the platform.
- Make sure to label each image accordingly, like Vatsal, Manoj, and Vijay.



Impulse Design

Once our dataset is uploaded, we'll design an Impulse (model pipeline) in Edge Impulse.

Choosing the Model Architecture

Model Choice: Convolutional Neural Network (CNN).
For image-related tasks like face recognition, CNNs are typically the best choice. CNNs can detect patterns (like facial features) and differentiate between individuals.

- Click on the Impulse Design tab in Edge Impulse.
- Choose the Image block. This will preprocess the images (resize, convert to grayscale if needed, etc.) and prepare them for model input.
- Select Image Classification as the task.
- Choose a CNN architecture for the project.
- We may need to resize our input images to a 320 x 320 pixels resolution to make the model efficient.

Impulse Design

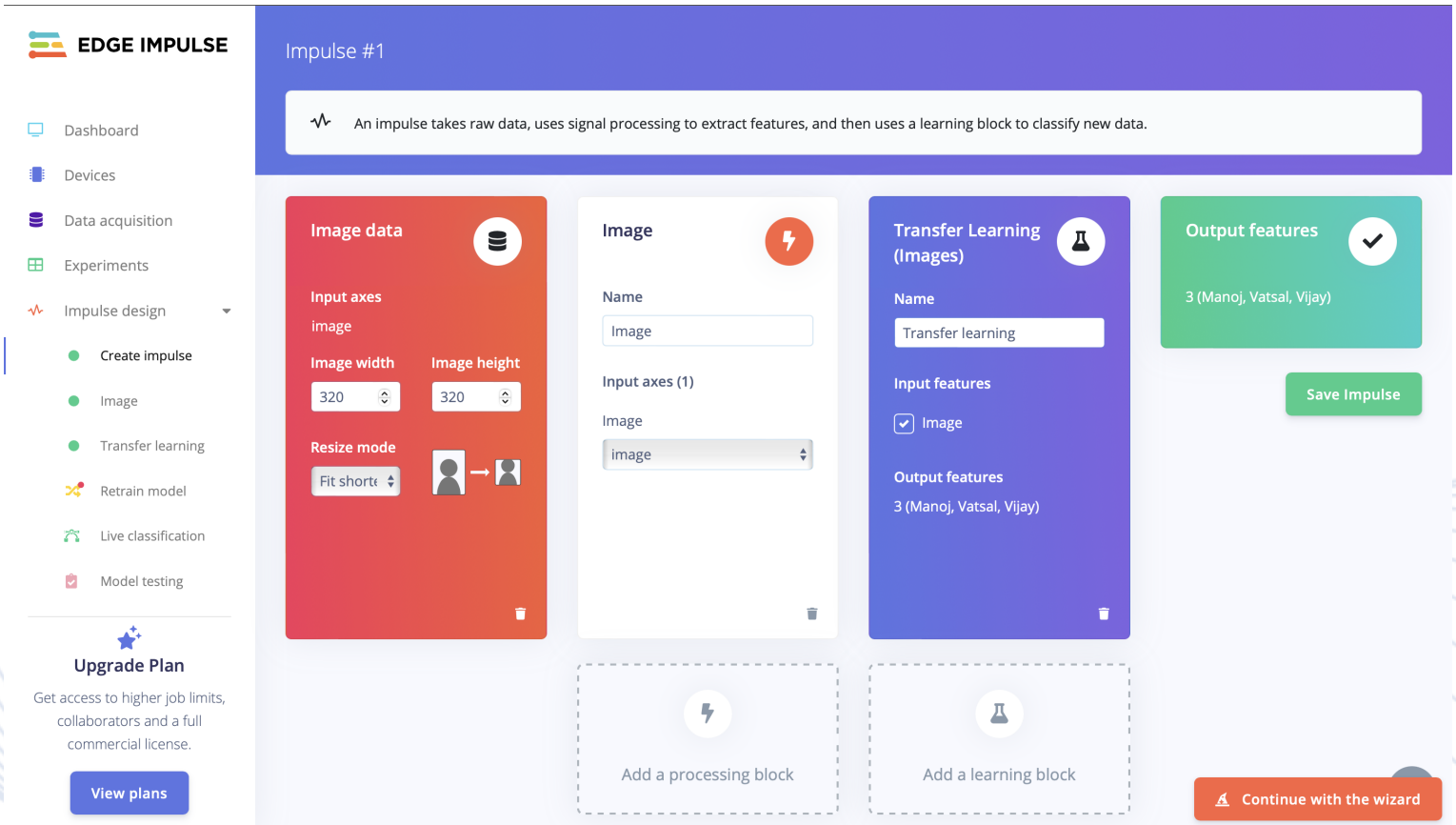


Figure 3: Impulse Design

Model Training

Configuring Training Parameters

- **Training/Validation Split:** Usually 80% training, 20% validation. But in our case, it's **76%** training, **24%** validation.
- **Number of training cycles (Epochs):** 25
- **Learning rate:** 0.0005

Model Training

- **Accuracy:** 83.3%
- **Loss:** 0.22

Model Training

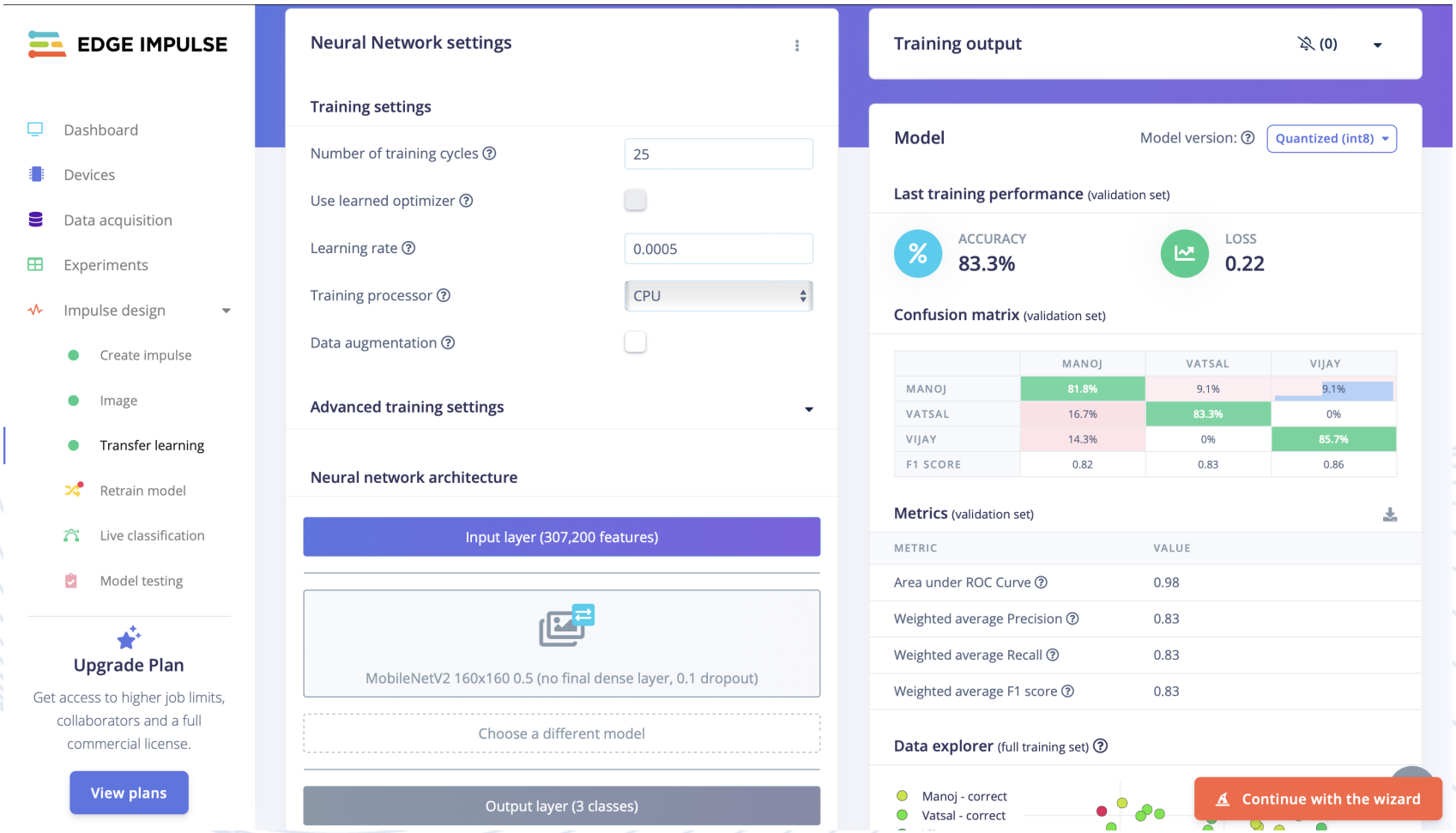
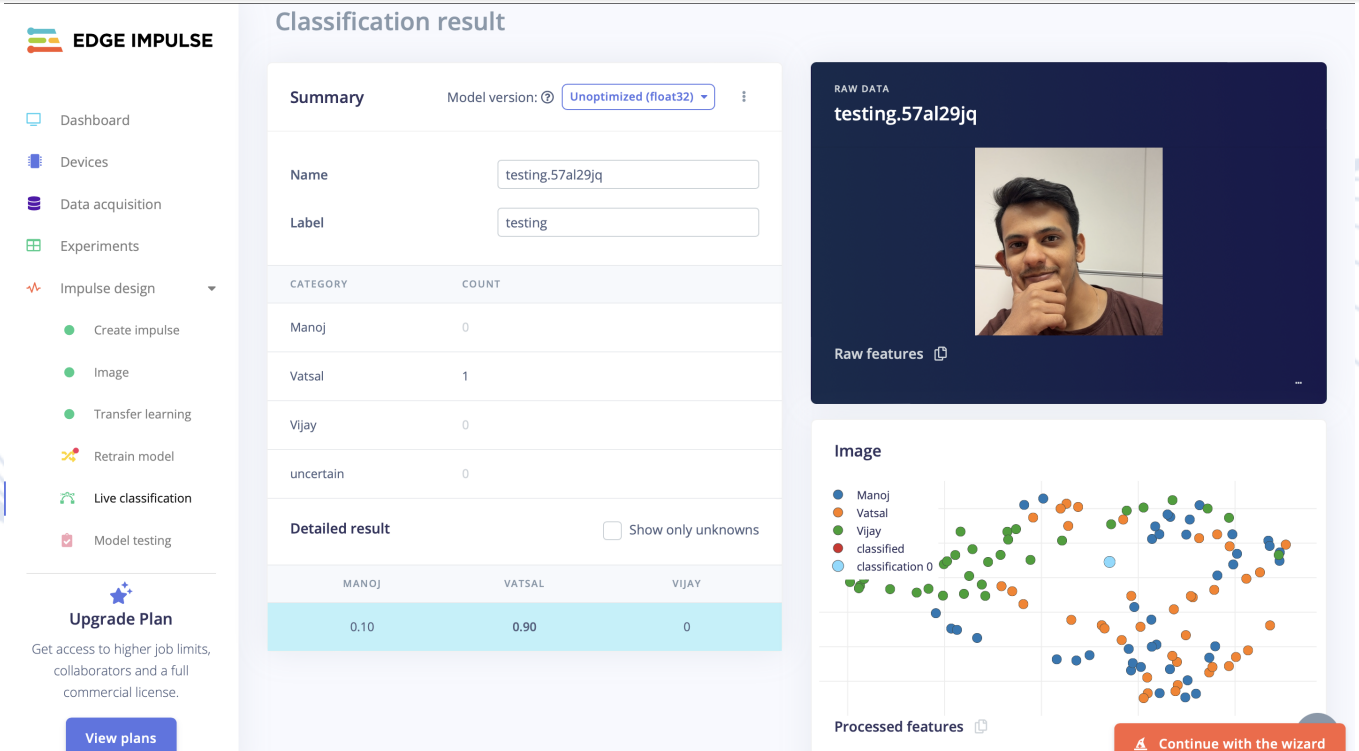


Figure 4: Model Training Progress

Model Testing

Evaluating the Model

- **To Test the model:** Go to "Live Classification" tab. we uploaded some new images and the model recognizes faces.
- **Confidence Score:** The system will show confidence scores for each class (Vatsal, Manoj, Vijay).



Model Deployment

Evaluating the Model

- Go to the Deployment tab.
- Select **TensorFlow Lite (.tflite)** as the export format.
- **Quantization:** Choose **int8 Quantization** if you plan to run the model on embedded systems like Portenta H7. Quantization reduces the model size and speeds up inference.

Thank you

for your attention