Katherine Chen

- Discussion -

[Part 1]

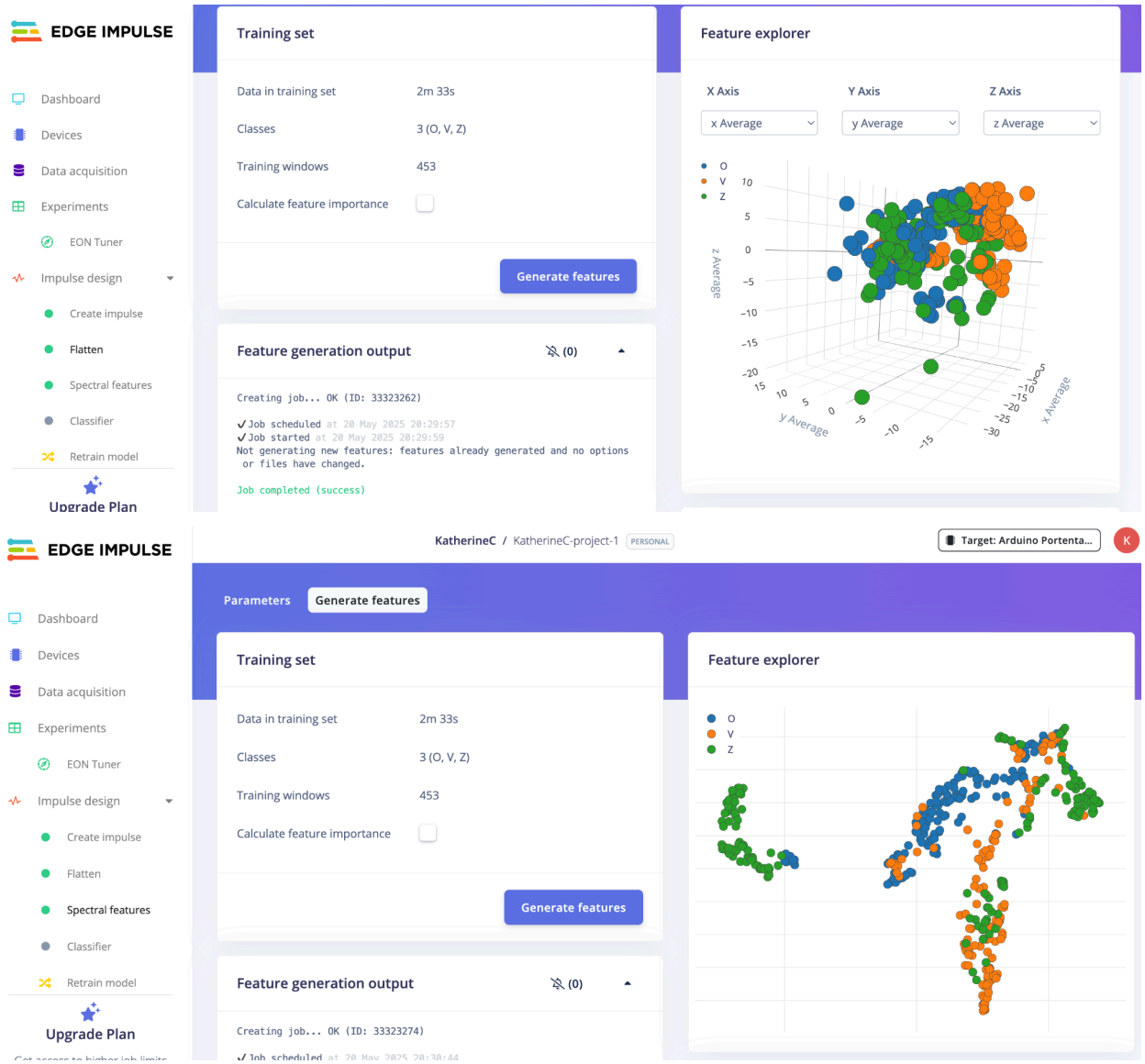
Using training data collected by multiple students makes the model more effective and reliable. In terms of effectiveness, it reduces the burden of data collection. Instead of collecting a large amount of data by myself, I can use a smaller amount from each person to build a more diverse dataset. For reliability, the model becomes more accurate when trained on data from different users. For example, when I trained the model using only my own data, the accuracy was around 40.97%. After including data from my classmates, the accuracy improved to about 60%. It also predicted my own gestures more accurately, because the model had learned more varied drawing styles.

[Part 2]

(2)

- For my impulse, I chose Flatten and Spectral Analysis as the processing blocks. The Flatten block helps convert the raw time-series data (x, y, z accelerometer readings) into a simple array format that is easier for machine learning models to process. It works well for gesture classification tasks where temporal order is less important than the overall movement pattern. The Spectral Analysis block extracts frequency-domain features from the accelerometer data. This helps the model capture repeating patterns or the "shape" of the gesture over time, especially useful for gestures that have distinct movement speeds or vibration patterns (like O vs. V). Combining both blocks allows the model to consider both time-domain and frequency-domain information, improving its ability to recognize subtle differences between gestures.
- The window size plays an important role in how the model processes and learns from the data. In my case, I used a window size of 500 milliseconds, a stride of 250 milliseconds, and a sampling frequency of 100Hz, collecting data from the x, y, and z axes. This configuration results in 50 data points per axis for each window, which means the input layer of the model receives 150 features per sample. I collected at least 60 samples for each of the three gesture labels: Z, V, and O, which provided a well-balanced and sufficient dataset for training.
- Using a 500ms window allowed the model to capture most gestures completely, especially slower ones like drawing the letter “O.” At the same time, the 250ms stride helped generate more training samples from the same data by overlapping windows, increasing training efficiency. A smaller window might have captured only part of a gesture and reduced accuracy, while a much larger window would have resulted in fewer samples and a more complex model. Overall, this window size and stride combination helped improve the model’s ability to recognize different drawing patterns more accurately and consistently.

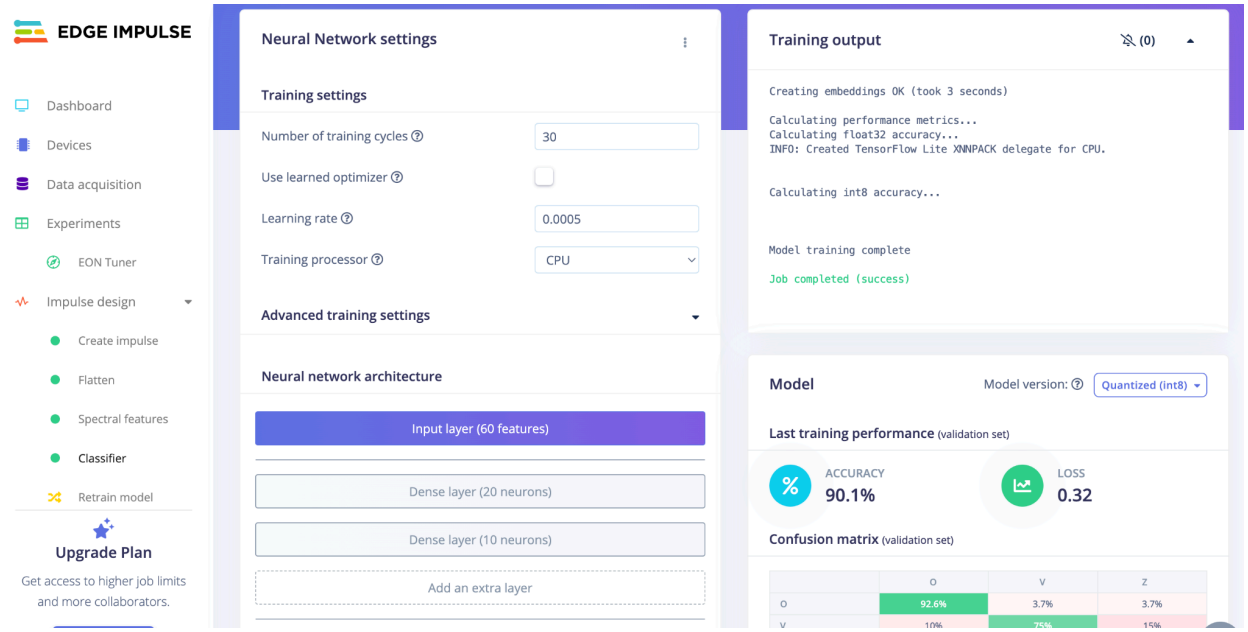
(3)



After generating features using Flatten and Spectral Analysis, I noticed that the data points show some level of grouping by class, although the separation between the three labels (Z, V, and O) is not very distinct. There are areas where the classes seem to overlap, but I can still roughly imagine where decision boundaries might go.

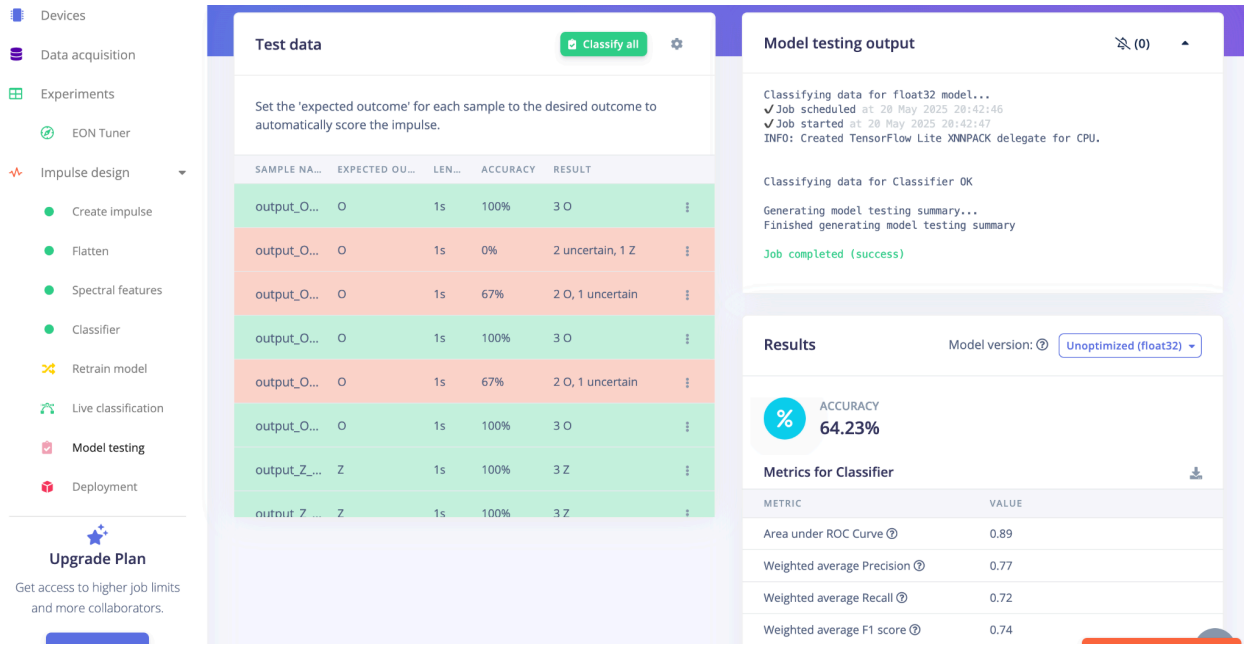
This suggests that the current DSP settings are capturing some useful patterns, but there may be room for improvement. For example, further tuning the window size, overlap, or exploring different DSP blocks like Spectrogram might help separate the features more clearly.

(4)



For my learning block, I selected the Classification option. I trained the model using 30 training cycles and a learning rate of 0.0005 on the CPU. The input layer receives 60 features extracted from the DSP block. I used the default neural network architecture provided by Edge Impulse for classification tasks. After training, the model achieved 90.1% accuracy and a loss of 0.32, which shows that the model performs well on the training data and has learned to distinguish the gestures effectively.

(5)



(7)

To further improve my model's performance, there are several strategies I could explore. First, I can collect more training data from more students. This would help the model generalize better by capturing more variations in how different people perform the same gestures. It would likely reduce misclassification caused by differences in drawing style or speed.

Second, I could experiment with different DSP blocks. Currently, I'm using Flatten and Spectral Analysis, but trying a Spectrogram might help capture more detailed frequency-based features, especially for complex or slower gestures like "O." These additional features could give the classifier a better understanding of motion patterns over time.

Another option would be to try a different classifier model. Right now I'm using a decision tree, which is simple and fast, but switching to a model like K-Nearest Neighbors or even a lightweight neural network could improve accuracy, especially if I have more training data and higher-dimensional features.

[Part 3]

(2)

I deployed the model to my ESP32 and tested real-time gestures using the wand.ino sketch. I performed 5 trials each for the gestures Z, V, and O. Out of 15 total tests, 9 predictions were correct, resulting in an overall accuracy of 60%. The model consistently recognized V and O with high confidence, but occasionally misclassified Z as V. Despite this, the wand responded quickly after each gesture. These results show that the model performs reliably in real-time scenarios, though further tuning or more data may help improve its accuracy with similar-looking gestures.

(6)

Link:
https://drive.google.com/drive/folders/1S-edyKTrTb8mWj_NRgr12DSsd1XiNJoZ?usp=sharing