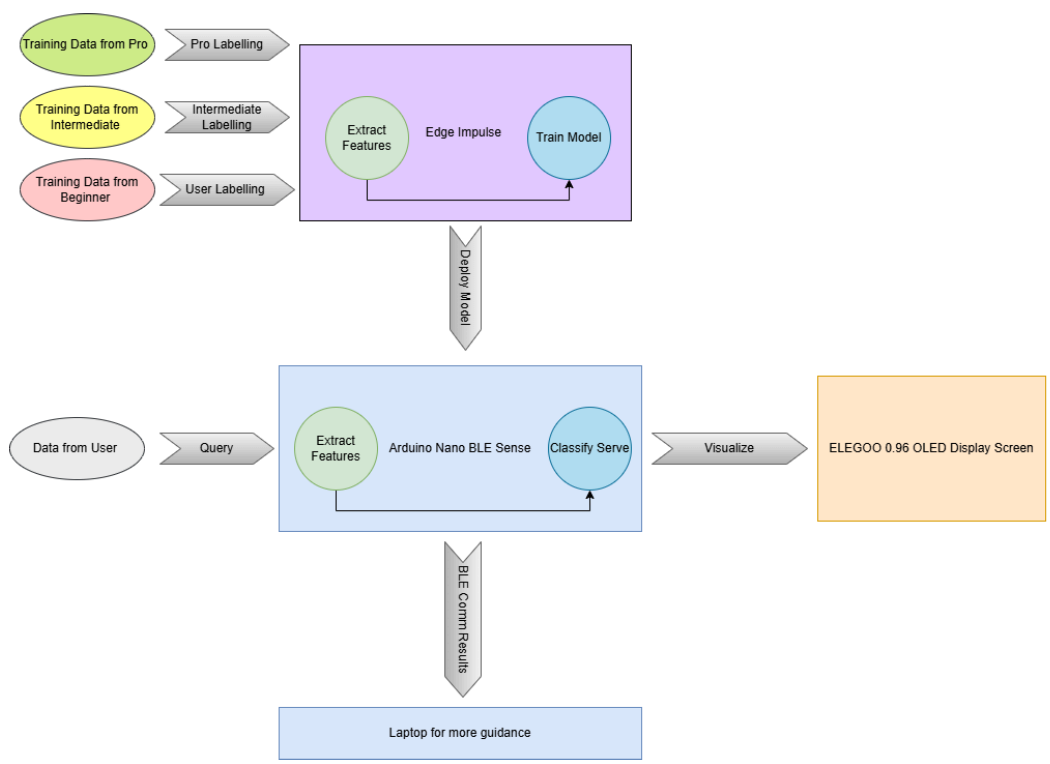
PocketPro

Brendan Wilhelm (bwilhelm), Kobe Zhang (kwzhang)

Abstract

Self-learning anything can be challenging. Our device, the PocketPro, is an embedded device that can be attached to a glove to allow table tennis enthusiasts and beginners to conveniently train their technique in practice by utilizing motion classification to classify and compare a user's actions with professional data. The PocketPro will also be displaying their serve data via an accessible screen. With this comparison, we will generate actionable insights for the user to improve their gameplay.

High-Level Block Diagram

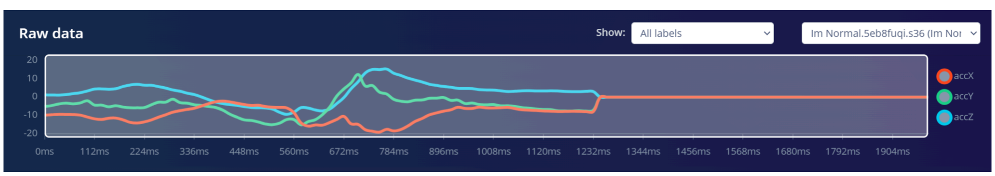


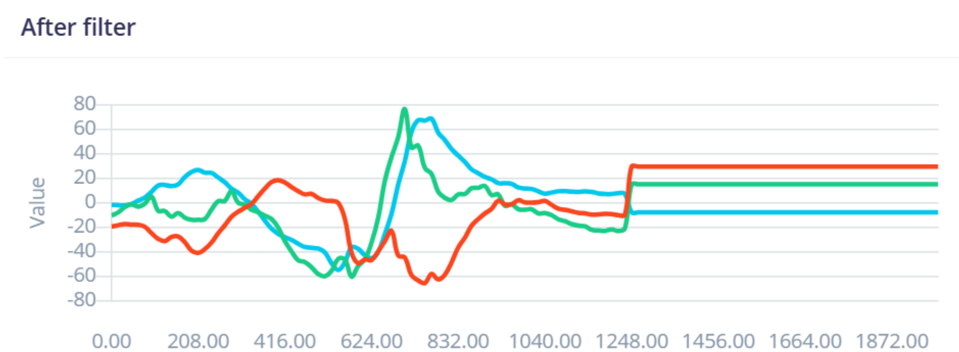
Dataset Source

Our method of data collection utilized the recording feature on Edge Impulse. We used the flash_mac.command script we were provided in class to flash the Arduino Nano BLE Sense with firmware which was compatible with Edge Impulse. Then using Google, we utilized the WebUSB feature which allowed us to directly record data to Edge Impulse. We began by instructing each player on the serve they were to perform and allowed them a few practice swings, and then began recording. We recorded for 80 seconds while they performed each serve, and used the automatic split feature to dice up the serve data into 1.25 second samples. We cleaned the data by hand by briefly scrolling through the collected serves, adjusting any samples as necessary to better align within the window. Based on our knowledge and experience playing table tennis, we then classified the recorded serves as beginner or intermediate. Unfortunately we were unable to obtain enough advanced data to have reliable classification for this category. The three different serves that each player performed were a normal non-spin serve, a chop/backspin serve, and a side spin serve which was performed by moving the paddle to the left on contact with the ball. We also collected idle data at various angles and let Edge Impulse automatically split the multiple 80 second intervals during training. Overall we collected around 1100 serves from roughly 15 individuals, with each individual collecting data for 10-15 minutes.

Feature Extraction

Our impulse utilized three different spectral analysis blocks based on our testing with serve data from the different serves. During testing and classification we noticed that there were a lot of similarities between the chop and side spin serve since the sensor is stationary on the player's finger, which shows similar acceleration data. To combat this the three spectral analysis blocks processed acceleration and gyroscope, acceleration, and magnetometer data respectively. We believe that the magnetometer and acceleration spectral analysis was useful in discerning between beginner and intermediate serves as the intermediate serves were often quicker with more acceleration than beginner serves. We also saw a difference in gyroscope data between chop and sidespin serves, which were difficult to discern. Using the spectral analysis blocks allowed us to scale up these features to be more important to the classification model.





Classification Architecture

Our classifier took in 252 features from the previous spectral analysis preprocessing blocks, which then fed into two dense layers. The dense layers had 80 neurons and 20 neurons respectively. We found after optimizing that due to the large amount of feature extraction done in the spectral preprocessing blocks that we could utilize a relatively lightweight deep neural network to go the final mile to classification.

Deployment Method

Our deployment method utilized the Arduino Nano BLE Sense with an ELEGOO 0.96 OLED display which was mounted on the player's hand. Originally we intended to deploy the sensor directly into the handle of the paddle, but to preserve our paddles we opted to deploy the Arduino on the forefinger of the player, with the display sitting on the back of their hand. We also intended to mount the microcontroller and display onto a glove with a dedicated battery, but last minute battery failures forced us to rely on a backup 15 foot power cable for the demonstration.

We utilized a software architecture which had a foreground thread handling data collection from the IMU, while a backup thread performed the classification and communication with the OLED and over Bluetooth to the laptop. We performed smoothing over 4 frames and relied on a 2 frame vote for a confirmed prediction. Once we saw a change in prediction from the previous prediction, we displayed X, Y, and Z axis acceleration data on the OLED for real time serve feedback. After the acceleration data was plotted, we then sent additional serve insights over Bluetooth using the Bleak package, such as maximum X and Z axis acceleration.

Confusion Matrix and Accuracy Metrics

Area under ROC Curve: 1.00

Weighted average Precision: 0.95

Weighted average Recall: 0.95

Weighted average F1 Score: 0.95

Note: Due to the consistency of the intermediate players and fewer players than the beginner categories, we saw extremely high accuracy.

Last training performance (validation set)

ACCURACY
95.4%

LOSS
0.11

Confusion matrix (validation set)

	BEG CHO	BEG NOR	BEG SIDE	IDLE	IM CHOP	IM NORM	IM SIDE S
BEG CHOP	93.6%	0%	6.4%	0%	0%	0%	0%
BEG NORMA	4.2%	95.8%	0%	0%	0%	0%	0%
BEG SIDE SP	6.7%	6.7%	86.7%	0%	0%	0%	0%
IDLE	0%	0%	1.9%	98.1%	0%	0%	0%
IM CHOP	0%	0%	0%	0%	100%	0%	0%
IM NORMAL	0%	0%	0%	0%	0%	100%	0%
IM SIDE SPI	0%	0%	0%	0%	0%	0%	100%
F1 SCORE	0.93	0.96	0.87	0.99	1.00	1.00	1.00

Challenges and Lessons

One of the biggest challenges of the project was data collection, since it was necessary to have a large amount of high quality serve data to train the model on. Luckily the dataset collection techniques we used to ensure consistency and data quality led us to achieve high training and testing validation accuracies. Even so, we were unable to collect enough data from the advanced category to have good prediction accuracies.

Another lesson we learned while working on this project was that it was incredibly important to replicate the conditions of the data gathering so we could achieve a higher accuracy. Additionally, we also saw that having different conditions for the usage of our device led to decreased accuracy during our demo. Because we used a small table as our demo table, servers were unable to put their usual amount of force into their serves, leading to false classifications.

Images and Videos

Images and videos were taken during the demonstration by Prof. Ziad.