Wearable sensing module for Table Tennis stroke detection

Sourin Ghosh
Department of Electrical Engineering
Indian Institute of Technology(IIT), Bombay
Mumbai, India
souringhosh1@live.com

Yogesh Gholap
Department of Electrical Engineering
IIT, Bombay
Mumbai, India
yogesh.gholap.2010@gmail.com

Siddharth Tallur
Department of Electrical Engineering
IIT, Bombay
Mumbai, India

2023 IEEE Applied Sensing Conference (APSCON) | 978-1-6654-6163-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/APSCON56343.2023.10101257

***Abstract*—Sensor-based analytics is a science of capturing the essentials of human motion and has been extensively used in sports by attaching external sensors to players for deriving meaningful insights, thereby improving game-playing quality. This paper presents the work towards developing a wearable Body Sensor Network(BSN) for Table Tennis(TT) stroke detection through self-learned features. By capturing acceleration and rotation parameters using Inertial Measurement Unit (IMU) sensors mounted on the player forearm and shoulder while playing a TT stroke, a neural network has been trained to predict forehand and backhand shots. The model runs on Arduino Nano 33 BLE and achieves a real-time prediction accuracy of 90.7%. Also, a memory-efficient Tensorflow-based Quantization Aware Training(QAT) model, with up to 4 times reduction in model size, has been developed to demonstrate comparable prediction accuracy with size reduction.**

***Index Terms*—Table tennis, sports analytics, IMU, body sensor, machine learning, neural network, tinyML, TensorFlow**

I. Introduction

Sport has come a long way from just being an activity of pleasure. Today, it is more than a billion-dollar industry [1]. There is always a learning curve to play few casual games or professional tournaments. Professional players hire a coach to learn the nuances and fine-tune their game [2]. Players need to focus on their physical fitness and proper technique to compete at the highest levels. The role of sports science and data analytics are important here. Analysis of player performance while playing these strokes is crucial for improving their strengths and weaknesses.

Wang et al. [3] discuss using data and simulations for tactical advantages in table tennis stroke placement, technique, and position. Table Tennis players are subjected to fitness tests consisting of alternate strokes of Forehand and Backhand after a specific time interval [4] [5]. One can deploy external cameras and body sensors to gather player performance data [6]. While such sensors capture the dynamics of the overall play, more insights can be gathered by mounting sensors on the player's body to capture acceleration and rotation signals of their movements [7]. A framework for capturing such signals is proposed by Sharma et al. [8] and uses IMU and audio sensors mounted on the player's body. Wearable sensors have also been proposed [9], [10] to capture stroke playing data and perform manual feature extraction to classify five different types of bat strokes. Blank et al. [11] propose a stroke classification design in which the IMU sensor is mounted on the table tennis bat and allows them to classify eight different types of strokes.

The key takeaway from prior art was using IMU sensors to measure a stroke quantitatively. The features needed to train an automated stroke classifier model always involve manual feature selection before feeding to a classifier for training. This motivated us to explore development of a wearable BSN to predict the Table Tennis player stroke (Fig. 1) without explicitly engineering features prior to classification but training a neural network only with IMU accelerometer and gyroscope signals [13].

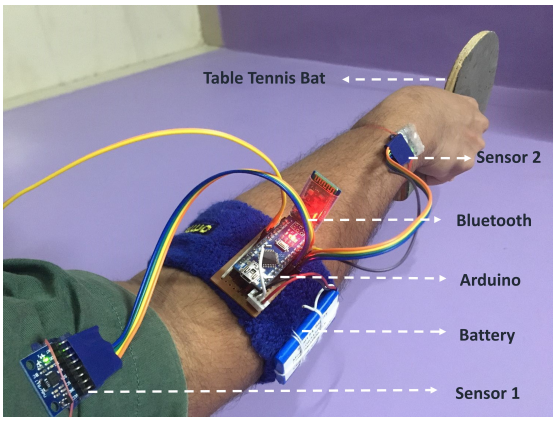

Fig. 1: Designed BSN module. One IMU sensor is attached to the forearm and another to the shoulder bicep and are connected to a microcontroller.

II. Methodology

The method applied towards classifying strokes are detailed in this section with the following inclusion criteria:

- Stroke data is captured from 5-6 people holding a TT bat in right arm shakehand grip.
- The designed system predicts forehand and backhand stroke motion only.

A. System Design

The entire system is divided into three stages: **Data Capture**, **Training** and **Deploy**, as shown in Fig. 2.

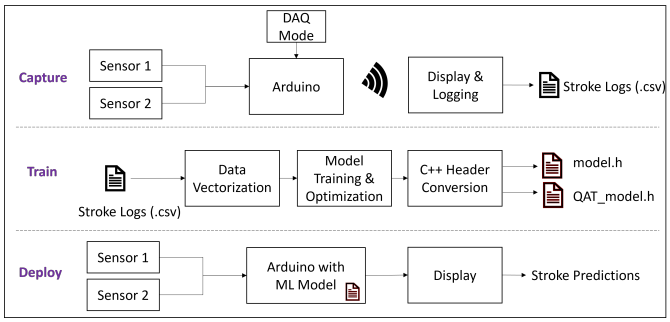

Fig. 2: System Diagram showing the Capture, Train and Deploy stages of TT Stroke classification

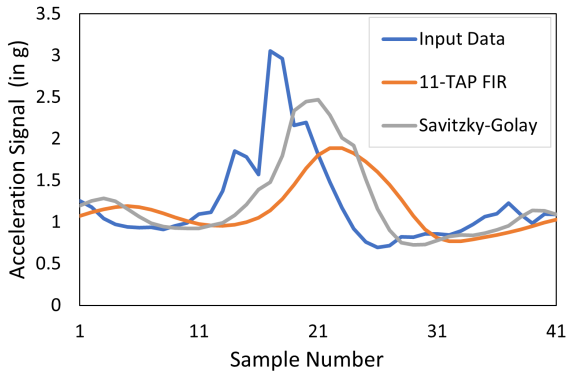

Fig. 3: Plot of accelerometer signal pattern on ball impact and corresponding filtered outputs

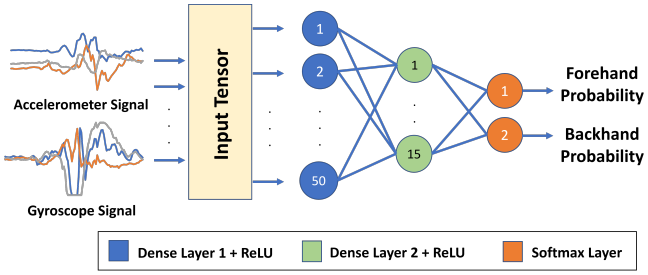

Fig. 4: Neural Network Architecture

The hardware setup used to **capture data** involves Arduino-Nano based prototype along with MPU-6050 IMU and HC-05 Bluetooth module. Secondly, the data acquired is then used for **training** a stroke classifier. In the third stage, the trained model is **deployed** on Arduino-Nano-BLE-33.

B. Hardware Design

Arduino-Nano is used for data acquisition, while Arduino-Nano-BLE-33 is used to deploy the trained classifier model. The Arduino-Nano-BLE-33 supports Tensorflow-lite library.

The prototypes designed for data acquisition and stroke inference were fixed on a sweatband to ease access.

C. Software Design

1) Data Logging:

- Two IMU sensors were placed on the arm, and their raw data was recorded. Six variables (Accelerometer A_x, A_y, A_z and Gyroscope G_x, Gy, G_z) per sensor were recorded, giving 12 variables to be analyzed.
- The data set is generated for three mutually exclusive stroke-motions (Stroke-Class), viz. Forehand, Backhand, and No-stroke. 750 stroke samples are recorded (250 forehand stroke samples, 250 backhand stroke samples, and 250 no-stroke samples). This sample dataset is used for training and testing the classifier.

2) Filtering ball impact spikes: The raw input data acquired from both the sensors is pre-processed to remove the outlier spike due to the impact of a ball on TT bat. To demonstrate deviations in signal envelope due to impact of a ball on TT bat, an Accelerometer vector A_{rms} is evaluated as,

$$A_{rms} = \sqrt{A_x^2 + A_y^2 + A_z^2} \tag{1}$$

Fig. 3 shows the accelerometer RMS ball impact signal and filtered outputs for forehand stroke motion. Ball impact on TT bat results in an impulse-shaped spike.

To filter the spike, two low pass filters: 11-tap Finite Impulse Response (FIR) and Savitzky-Golay(SG) moving average filter [12], has been implemented.

The 11-TAP FIR filter is designed with corner frequency $F_c = 2.5\ Hz$ since the stroke envelope has spike values with an average duration of 400 ms (20 samples) for each stroke sampled at 50 Hz. The filter coefficients are calculated for FIR and SG moving average filter [14].

3) Data window selection: A stroke period with approximately 70 samples representing the stroke duration of 1.5 seconds, is chosen. Since there are two sensors in the proposed design, and each sensor provides six-axis values, these twelve axes generate 70 samples each resulting in an input tensor of 840 samples. For every stroke motion in the dataset, this tensor is created to prepare the training set.

4) Data Normalization:

- As seen earlier, the sensor signals have different dimensions and ranges. It is essential to normalize them to a range of $0 - 1$ to represent the same scale and for the subsequent data processing algorithm to make sense of the information.
- To perform normalization on the accelerometer and gyroscope data, add the offset and divide the signal by the range.
- The described normalization technique was performed on the signal data frame. Note that our accelerometer data varies from ± 2 g and the gyroscope signal varies from $\pm 250\ ^\circ/s$.

5) Designing the binary classification neural network: The neural network created is highlighted in Fig. 4. The network is a three-layer dense network consisting of 50 fully connected neurons in the first layer 15 fully connected neurons in the second layer, followed by a softmax layer to predict class probability.

The total trainable parameters for this model are $42,847$. The model is run in Arduino as a C++ header file. In our case, the *model.h* file generated using the Tensorflow package is $\approx 173\ KB$ in size. Since embedded devices have limited storage, this model can be optimized with respect to the number of trainable parameters. This paper highlights an optimization method adopted to reduce the model size without compromising classification accuracy.

6) Memory-optimized Quantization Aware Training (QAT): Deploying neural networks on embedded devices such as Arduino-Nano-BLE-33 needs compact optimizations in model size without compromising on accuracy. Since the presented multi-layer neural network solves a non-linear problem, the model has a 32-bit floating-point representation. Interestingly, Tensorflow allows QAT Application Programming Interface (API) in their Model Optimization Toolkit and provides a choice to use lower precision(8-bit Integer). QAT enables to train and deploy models with performance and size benefits of quantization while retaining close to original accuracy. The QAT model learns parameters that are robust to quantization.

III. Results and Discussion

The model described in the previous section is written in python using Google Colab Pro+.

1) Comparing Non-QAT versus QAT model performance accuracy: The confusion matrix in Table Ia is generated by testing the non-QAT and QAT models on 50 Forehand and 50 Backhand strokes picked randomly from the train-test set. On this test data, our non-QAT model predicted strokes with 100% accuracy while the QAT model performed with an accuracy of 80% for the Forehand and 78% for Backhand stroke predictions(Table Ib).

TABLE I: Confusion Matrices (F=Forehand, B=Backhand) (a) Non-QAT Model (b) QAT Model

	F	B
F	**50**	0
B	0	**50**

(a)

	F	B
F	**40**	10
B	11	**39**

(b)

2) Comparing Non-QAT versus QAT memory occupancy: The Tensorflow-lite tinyML QAT model shows a **four-times reduction** in memory usage, as seen in Table IIa. Memory usage was reduced for both Tensorflow-lite and C++ converted model instances.

TABLE II: (a) Model size comparison (b) Accuracy comparison (real-time deployment)

Model	Size
TFLite tinyML	173 KB
TFLite QAT	**45 KB**
Non-QAT C++ Header File	1879 KB
QAT C++ Header File	**479 KB**

(a)

Model	Accuracy
Non-QAT	90.70%
QAT	**85.70%**

(b)

It is also seen that a $173KB$ Tensorflow-lite model file, when converted to a C++ header file, takes $1879KB$ memory. The increase is due to the lexical conversions of the model weight numbers into HEX values for C++. **The stroke inference accuracy during real-time testing, as seen from Table IIb, is almost comparable.**

IV. Conclusion

A real-time wearable module for Table Tennis stroke detection is presented in this work. By using a neural network to predict the strokes, it has been shown that a-priori feature extraction is not necessary before classification. It becomes useful in solving problems when the features are not visible just by observing the data. This work leverages Tenserflow-Lite APIs to be deployed on Arduino-nano-BLE-33 embedded setup mounted on player forearm for stroke prediction. The strokes played by a table tennis player are detected with over 90% accuracy. A memory-efficient QAT model was also implemented to achieve 85.7% accuracy, and a 75% reduction in memory requirement was achieved on target hardware. As future work, more complex strokes like top-spin, pen-hold grip, smash and push, can be detected with the help of additional sensors placed on the palm of the player.

References

[1] D. A. Heitner, "Money and sports: Economic realities of being an athlete,", *DePaul J. Sports L. & Contemp. Probs.*, vol 8, pp. 161, 2011.

[2] V. Trninić, V. Papić, M. Trninić, & Others, "Role of expert coaches in development of top-level athletes' careers in individual and team sports,", *Acta Kinesiologica*, vol 3, no 1, pp. 98–105, 2009.

[3] J. Wang et al., "Tac-simur: Tactic-based simulative visual analytics of table tennis,", *IEEE transactions on visualization and computer graphics*, vol 26, no 1, pp. 407–417, 2019.

[4] J. Lapszo, "The method of research into the speed of specific movements and anticipation in sport under simulated conditions in table tennis,", *Science and Racket Sports II*, pp. 135–142, 1998.

[5] J. Fallby, "21 Technical rehearsal and imagery: a system for enhancing technical skills in table tennis,", *Science and Racket Sports*, pp. 153, 2002.

[6] L. Draschkowitz, C. Draschkowitz, & H. Hlavacs, "Using video analysis and machine learning for predicting shot success in table tennis,", *EAI Endorsed Transactions on Creative Technologies*, vol 2, no 5, pp. e2, 2015.

[7] H. Harms, O. Amft, R. Winkler, J. Schumm, M. Kusserow, & G. Tröster, "Ethos: Miniature orientation sensor for wearable human motion analysis,", *SENSORS, 2010 IEEE*, 2010, pp. 1037–1042.

[8] M. Sharma, A. Anand, R. Srivastava, & L. Kaligounder, "Wearable audio and IMU based shot detection in racquet sports,", *arXiv preprint arXiv:1805. 05456*, 2018.

[9] R. Liu et al., "Table tennis stroke recognition based on body sensor network,", in *International Conference on Internet and Distributed Computing Systems*, 2019, pp. 1–10.

[10] M. Kos en I. Kramberger, "Tennis stroke consistency analysis using miniature wearable IMU,", in 2018 25th *International Conference on Systems, Signals and Image Processing* (IWSSIP), 2018, pp. 1–4.

[11] P. Blank, J. Hoßbach, D. Schuldhaus, & B. M. Eskofier, "Sensor-based stroke detection and stroke type classification in table tennis,", in *Proceedings of the 2015 ACM International Symposium on Wearable Computers*, 2015, pp. 93–100.

[12] R. W. Schafer, "What is a Savitzky-Golay filter?[lecture notes],", *IEEE Signal processing magazine*, vol 28, no 4, pp. 111–117, 2011.

[13] K. Dokic, T. Mesic, & M. Martinovic, "Table tennis forehand and backhand stroke recognition based on neural network,", *International Conference on Advances in Computing and Data Sciences*, 2020, pp. 24–35.

[14] A. Sorvala, E. Alasaarela, H. Sorvoja, en R. Myllylä, "A two-threshold fall detection algorithm for reducing false alarms,", *2012 6th International Symposium on Medical Information and Communication Technology (ISMICT)*, 2012, pp. 1–4.