Monitoring of a Padel match using a Smartwatch and UWB sensors

Tommaso Amarante, Iacopo Bicchierini, Matteo Guidotti, Edoardo Morucci
t.amarante@studenti.unipi.it, i.bicchierini@studenti.unipi.it, m.guidotti@studenti.unipi.it, e.morucci@studenti.unipi.it

https://github.com/EdoardoMorucci/MyPadel

ABSTRACT

Automatic Padel stroke recognition can be useful to Padel players to improve his training performances. We noticed that there is no application for monitoring an entire Padel match. We gathered raw data from a Padel match using gyroscope and accelerometer, we then investigated the stroke detection and the following classification of four different strokes. We trained a neural network from scratch, and we built an application with a smartwatch companion to collect data, moreover we used the data generated by UWB sensors together with the output from the neural network to construct a scatter plot so that the user can visualize the position of his strokes in the court.

1 Introduction

In literature detection and classification of a stroke are carried out using sophisticated IMU sensors or Computer Vision techniques[2]. Since the last is generally much expensive, we decided to exploit sensors. Our aim is to demonstrate whether a cheap smartwatch could be used to achieve the previously mentioned goal. In addition, we decided to use a set of UWB sensors to enrich the user experience, which helps the player to visualize the overall match performance knowing in which part of the court a particular stroke was performed. Since we did not find a proper dataset already created, we decided to gather information with a simple application on an Android smartwatch.

2 Architecture

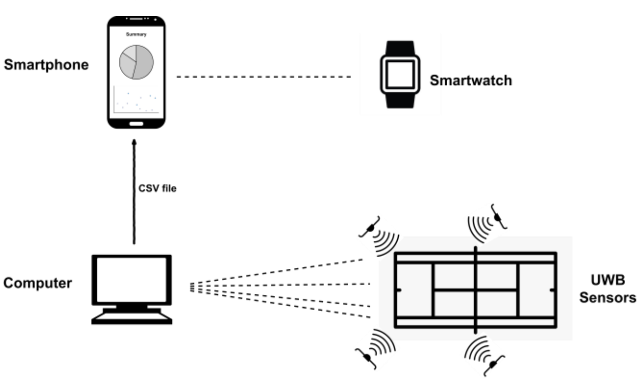

Figure 1 - Architecture

Our infrastructure is made up of a smartwatch that, thanks to an WearOS application, communicates via Bluetooth with the smartphone application sending every 5 seconds the gyroscope and accelerometer values. The other source of data is the UWB sensors. We put the four anchors in the angles of the paddle court, then we put the node sensor on during the match. In order to obtain a log file, we connected the listener to a laptop and this helped us construct a log for drawing the scatter plot. Since nowadays the majority of the smartphones are not equipped with a UWB receiver, we manually put the log file in the filesystem of our smartphone to process data. The whole architecture is shown in ***Figure 1***.
The core effort in our project is the development of the Android application that processes the two sources of data resulting in a user-friendly interface. The application stores the summary of all the matches performed by the user, which will be later discussed.

2.1 Data Collection and Manipulation

Since we did not find suitable dataset, we managed to create our own ones. For every type of stroke, we gathered 30 minutes of continuous shots, we rotated in this operation to increase the variance of the stroke signal.

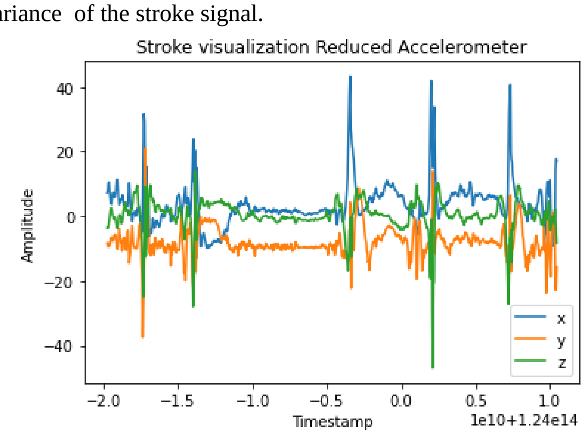

Figure 2 – 30 seconds of signal after frequency reduction

After the data collection phase, we were worried about the possibility to classify the stroke, because of the limited capacity of the sensors of the smartwatch with respect to the IMU ones. Plotting the data, it results that the range of sensing of both sensors was comparable to the more sophisticated ones.
We then smoothed the signal. The problem that came out was the too high frequency of sampling, 100 Hz. Since our idea was to use

a simple NN as classifier, we would like to keep the input not so huge, so we reduced the sampling rate, making the average of 4 consecutive samples. We got a frequency of 25Hz and a consequent feature set of 150 values.

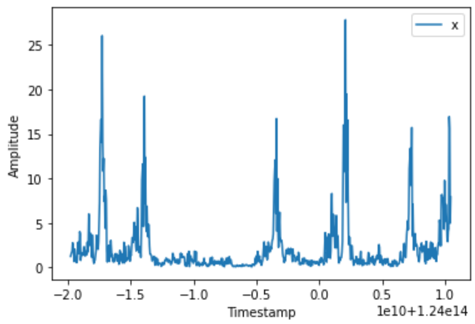

Figure 3 – Average derivative of the 3 axis of accelerometer

The next phase was the Stroke Detection and, as suggested in the paper [1], we used a threshold on the average derivative of the 3 axis related to the accelerometer. Plotting the signal it was clear that a threshold of 10 would give the best results as shown in the ***Figure 3***, then in order to avoid false positives, we set a minimum of 1.2 second time between two detectable strokes.

The features related to a single shot are the raw values of the sensors in a window of 1 second centered in the maximum derivative timestamp.

To be sure that the classifier would be able to recognize the strokes, we gave a look at the signal, which was as expected clearly recognizable, as shown in ***Figure 4***.

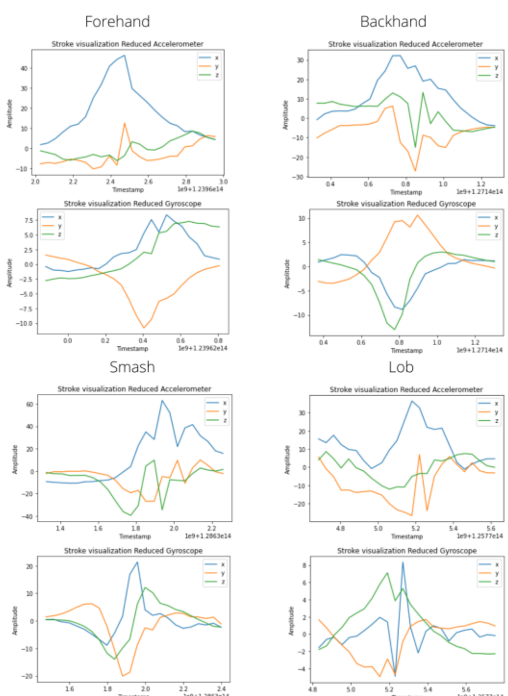

Figure 4 – Example of signal for each stroke type

2.2 Data Classification with Neural Network

After gathering all the information from the signals, we constructed both the training set and the test set with a percentage of 70% - 30%. After that we built a Neural Network model from scratch in Python with Tensorflow. The input of our Neural Network is composed of 150 raw values from a window of 1 second, as each axis of the two sensors has 25 samples. The network has 2 dense layers, the first one uses the Relu as activation function while the second one Softmax function with Adam as optimizer. We set the learning rate to 0.001 and the number of epochs to 25, this gave us a model with an accuracy of 0.9674 on the training set and an accuracy of 0.7236 on the test set.

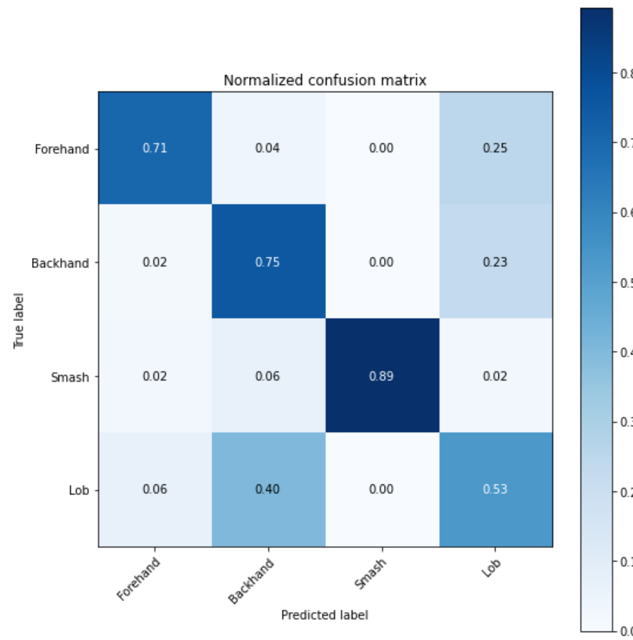

Figure 5 - Confusion Matrix

The confusion matrix obtained by the model is shown in ***Figure 5***, as we can see we got good results for all the strokes, the only issue was the classification of the lob, which might be caused by a wrong execution of the stroke during the data collection phase, as already mentioned in [2].

2.3 Positioning

To gather positioning information during the Padel activity we used Decawave MDEK1001 Kit. We deployed 6 devices, one acted as listener connected with USB to a PC, 4 had the role of computing the position and were attached to the corners of the walls of the Padel court at a height of 1.5m, the remaining one was the mobile node. The listener node gathered information from the network and generated a log with the position in x,y,z and a value related to the accuracy of the positioning.

This log information, crossed with the one resulting from the NN, allowed us to give a deeper perspective related to the Padel activities, as shown in the experimental result paragraph.

3 Experimental results and Application

A brief overview about the UI of the Android application is useful in order to better highlight the experimental results.

3.1 UI

Our application has three sections. The *Profile* is conceived to gather the user information, which could be later on exploited as an additional input parameter for the training of the NN. The *Activity* section shows only the chronometer related to the current session, which is started using the specific button in the smartwatch application.

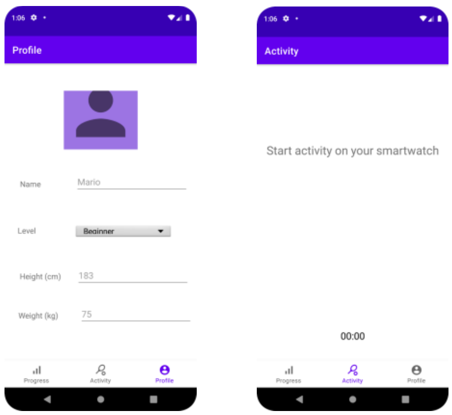

Figure 6 - Profile and Activity sections

The *Progress* section contains the information related to all the previous Padel sessions, chronologically arranged.

3.2 Results

We tested whether our system worked out during some Padel sessions. It performed optimally in providing useful information about the Padel session. In particular, it returned the stroke type distribution and the position in which each strokes was performed, as shown in ***Figure 7***.

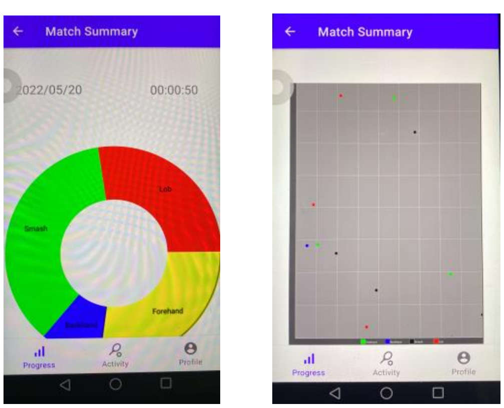

Figure 7 – Match Summary

4 Conclusion

In conclusion our initial aim can be considered achieved since, using cheap hardware and pretty simple software, we are able to give useful information regarding the activity of a Padel player.

REFERENCES

[1] Christopher J. Ebner and Rainhard Dieter Findling. Tennis Stroke Classification: ComparingWrist and Racket as IMU Sensor Position. MoMM2019: Proceedings of the 17th International Conference on Advances in Mobile Computing & Multimedia, December 2019, Pages 74–83 DOI: https://doi.org/10.1145/3365921.3365929

[2] Luis Pardo and David Perez and Carlos Urunuela, *Detection of Tennis Activities with Wearable Sensors,* Sensors 2019, 19, 5004, DOI: https://doi.org/10.3390/s19225004