Training Smarter with OpenEarable: A Boxing Gesture Recognition Dashboard Integration

Thomas Sepanosian*
University of Twente
Enschede, the Netherlands
t.sepanosian@student.utwente.nl

Ozlem Durmaz Incel
University of Twente
Pervasive Systems Research Group
Enschede, the Netherlands
ozlem.durmaz@utwente.nl

Figure 1: Augmenting the OpenEarable Platform by the Recognition of Boxing Head Gestures

Abstract

Earables, wearable devices worn around the ear, offer new possibilities for sports applications requiring precise head movement analysis, such as boxing. However, boxing-specific gesture recognition using IMU sensors integrated into earables remains underexplored. This work addresses this gap by investigating the potential of the open-source *OpenEarable* platform for real-time recognition of defensive boxing manoeuvres, including slipping, rolling and pulling back. We propose an extension to *OpenEarable*, integrating a Python server that leverages machine learning and dynamic time warping for gesture recognition. Furthermore, the web dashboard is enhanced to enable server communication and implement a gesture mirroring feature, providing real-time visual feedback. Real-time testing achieved a high accuracy of 96%, with feedback delivered within one second. All the system components are made available in a GitHub repository.

CCS Concepts

• **Human-centered computing → Ubiquitous and mobile computing systems and tools**.

Keywords

Earables, Inertial Measurement Unit, Real-time Head Gesture Recognition

*Corresponding author.

UbiComp Companion '24, October 5–9, 2024, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-1058-2/24/10
https://doi.org/10.1145/3675094.3678481

ACM Reference Format:
Thomas Sepanosian and Ozlem Durmaz Incel. 2024. Training Smarter with OpenEarable: A Boxing Gesture Recognition Dashboard Integration. In *Companion of the 2024 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp Companion '24), October 5–9, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3675094.3678481

1 Introduction

Earables are wearable devices specifically designed for placement in or around the ear. These devices offer advantages over traditional earphones by incorporating a suite of sensors, enabling a range of potential applications [6]. For example, they are utilized for sensing fine-grained facial expressions [9], for user authentication by detecting heart rate, gait, and breathing patterns [1]. Their placement on the head makes them particularly well-suited for tasks requiring head gesture recognition, as demonstrated by Montanari et al., who employed PPG signals for this purpose [3].

Head gestures have an important role in boxing, particularly in defensive maneuvers. Earables' real-time recognition capabilities could significantly benefit boxing training by also providing immediate feedback. To the best of our knowledge, they have not been utilized in boxing-specific gesture recognition. This research aims to address this gap by investigating the suitability of the open-source *OpenEarable* platform [7] for real-time recognition of boxing head gestures.

The *OpenEarable* platform is an entirely open-source initiative [7], granting researchers and developers the ability to directly modify and improve both its hardware and software components. This open-source nature enables rapid prototyping, making it an ideal platform for investigating applications in sensor-based gesture recognition.

This paper addresses the following research question: *How can a boxing head gesture recognition system be integrated into the OpenEarable framework to provide real-time feedback?* For recognition, we propose integrating classical machine learning algorithms and dynamic time warping (DTW). While the details of the model training and testing for the gesture recognition part are presented in a separate paper currently under peer review, in this paper, we focus on integrating the recognition model to the *OpenEarable* dashboard and providing feedback to the user through gesture mirroring over the dashboard.

The remainder of this paper is organized as follows: Section 2 reviews the *OpenEarable* platform. Section 3 outlines the methods used in this study. Finally, Section 4 draws conclusions and discusses potential avenues for future work.

2 Background: OpenEarable Dashboard and Applications

The *OpenEarable* platform offers a collection of open-source software and hardware components for ear-based wearable application development. One of its software components includes a Flutter library, which interfaces directly with the hardware and supports features such as showing live data alongside several applications.

These applications include a `Jump Height Test`, `Jump Rope Counter`, and a `Posture Tracker` [1]. Each application employs specific preprocessing techniques, such as applying a Kalman filter. Primarily, activity recognition within these tools is managed through threshold-based algorithms. For example, the `Jump Rope Counter` identifies jumps by detecting when acceleration magnitude exceeds a predetermined threshold. While effective or basic tasks, this method might limit the potential to recognize more complex gestures, which might not conform to predefined thresholds and, therefore, could benefit from alternative approaches. Furthermore, the platform's web dashboard, primarily used for data retrieval, lacks the interactive tools of its Flutter counterpart.

This paper bridges these gaps by augmenting the *OpenEarable* platform's dashboard. Specifically, our augmentation aims to provide real-time boxing head-gesture recognition feedback to users. This advancement will allow for integrating more complex recognition systems, including machine learning algorithms and dynamic time warping, thus broadening the scope of potential applications.

3 Methodology

This section outlines the methodologies employed to address our research question concerning real-time boxing gesture recognition using the *OpenEarable* platform. Figure 2 provides a high-level overview of the research methodology. The following subsections provide detailed information regarding the blocks in the overview.

3.1 Server Implementation

An application separate from the primary dashboard, referred to as the `Server` has been developed in Python to handle gesture recognition[2]. This choice of Python was driven by the vast array of libraries and support for machine learning applications, allowing for the rapid development of gesture recognition systems.

[1]As showcased on: https://github.com/OpenEarable/app, accessed in June 2024
[2]Available at: https://github.com/Thomas-mp4/EarableBoxingHeadGestureRecognition

3.1.1 Data Collection & Preprocessing: Gestures. We focused on three fundamental defensive manoeuvres employed in boxing: slipping, rolling and pulling back. These are important gestures in defence, enabling the boxers to evade incoming punches and establish advantageous counterattack positions:

- **Slipping**: This manoeuvre involves quick lateral movement of the head to either side, approximately the width of a boxing glove, to avoid straight punches aimed at the head. It is typically used to counter jabs and crosses, allowing for immediate counterattacks due to the minimal movement required. This is simulated by performing the slip in front of a boxing glove attached to the roof at head-level height, akin to a slip bag.
- **Rolling**: Also known as bobbing and weaving, rolling involves moving the head in a circular "U" shape motion. This technique is crucial for dodging hooks and uppercuts. The study simulates this by performing rolls underneath the slip-bag set at head height, encouraging replicable and consistent rolls.
- **Pulling Back**: Also known as the lean back, involves a quick backward movement of the head to avoid straight punches. Similar to the slip, for this study, pull-backs are performed in front of a slip bag.

Figure 3 provides a detailed illustration of the gestures. Given the variability in boxing styles, this study standardizes the maneuvers by using an orthodox stance, characterized by the left foot forward and hands guarding the face and torso. Approximately 460 repetitions have been collected, per gesture, over the span of multiple sessions, by a single hobby-level boxer.

3.1.2 Model Development. The *OpenEarable* dashboard is a web-dashboard, and thus utilizes JavaScript for functionalities. This adds to the system's overall compatibility and ease of use, as utilization of the dashboard does not require any elaborate setup apart from deployment. To expand the platform's capabilities, we integrated a dedicated Python server. This allows for the implementation of tools not natively supported by JavaScript.

We utilized the `SciKit Learn` library [4] to employ classical machine learning algorithms such as the `Random Forest`. Additionally, for more nuanced gesture recognition, we employed `fastdtw` [8] in conjunction with `Dynamic Time Warping Barycenter Averaging` [5]. With this system, we achieved an accuracy of 99% on a 70:30 training to testing data split. In real-world, live testing, with 50 repetitions per gesture, the system maintained a high accuracy of 96%. This demonstrates the efficacy of our integrated approach, providing a foundation for further enhancements. As mentioned, the details of the models for recognition were discussed in another paper which is currently under review.

3.1.3 RESTful API. To effectively deploy our Python server for utilization by the *OpenEarable* dashboard, we employed `Flask` [2], a flexible and lightweight web application framework. We set up a RESTful API on the server, primarily using a simple POST-request endpoint. This endpoint accepts an input sequence in JSON format from the client side, after which it utilizes the models and returns a prediction. For model integration, we utilized Python's `pickle` module to serialize our pre-trained gesture recognition models.

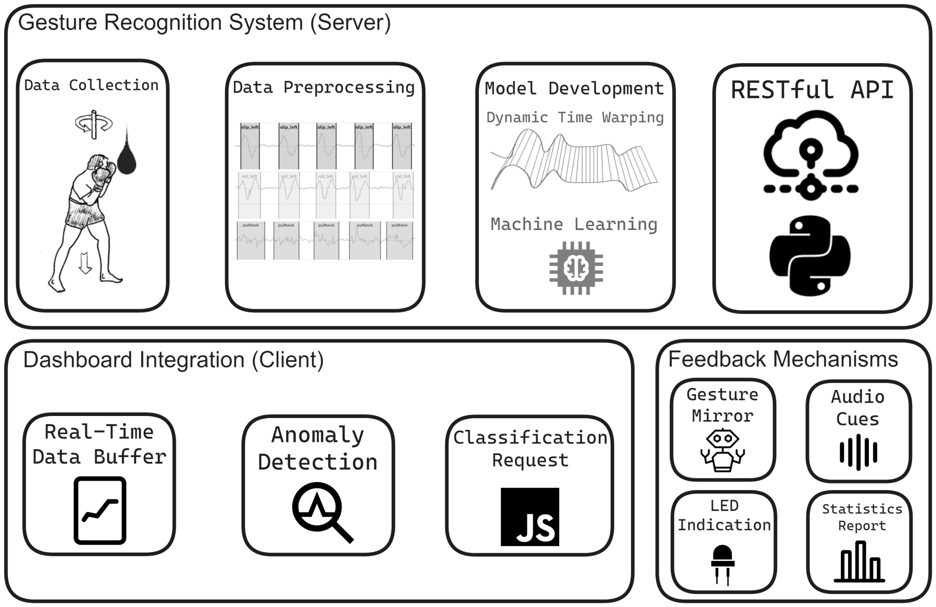

Figure 2: System Overview

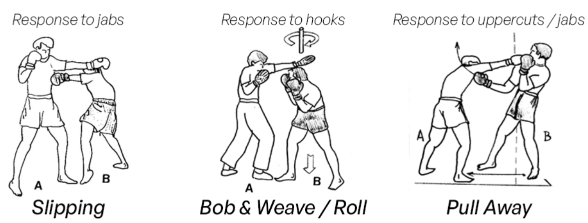

Figure 3: The target head gestures demonstrated by practitioner B, sparring against opponent A

3.2 Client Implementation

We utilize the *OpenEarable's* web-dashboard to communicate with the earable and to provide gesture recognition feedback [3]. We refer to the augmented dashboard as the `Client` [4].

The core functionality of the client in the gesture recognition system is to retrieve sensor data, detect anomalies, and obtain feedback through communication with the server. It achieves this by integrating with the dashboard's `sensorManager` to collect all recent data in a buffer. This buffer is monitored for anomalies, indicating potential gestures. Upon detection of a complete gesture, the data is sent to the server via a POST request, where the data is processed, the gesture is identified, and the result is returned to the dashboard.

[3]Available at: https://github.com/OpenEarable/dashboard, accessed in June 2024
[4]Available at: https://github.com/Thomas-mp4/OpenEarableAugmentedDashboard-BoxingHeadGestureRecognition

3.3 Feedback Mechanisms

There are multiple ways to convey recognition results back to the user. With the *OpenEarable* platform, it is possible to adjust the LED indicator, play certain audio cues, or summarize gesture data over a period to generate a report. In our implementation, we opted for visual feedback through a feature we term the gesture mirror. This mirror utilizes animated character models to visually represent recognized gestures. These animations were created using Adobe Mixamo[5], which provided pre-rigged models and animations. Further customizations were applied using Blender[6], which allowed the models to be exported as GL Transmission Format Binary files (.glb), a format supporting 3D-assets. These models are showcased in Figure 5. These models were implemented in the dashboard through the use of `model-viewer`[7]. Figure 4 showcases the augmented dashboard, with the recognition of a right roll, including the highlighting of a label indicator and the gesture mirror performing the gesture. Inspecting a video recording of the augmented *OpenEarable* dashboard, recorded at 60 frames per second, predictions are approximately made 50 frames after the anomaly ends (ignoring the moment the anomaly has been recognized as ending).

4 Conclusion

This study contributed to the augmentation of the open-source *OpenEarable* platform to achieve real-time boxing head-gesture recognition. The gestures of interest included left/right slips, left/right rolls, and pullbacks. Our approach consisted of developing a dedicated Python server, hosting a system capable of recognizing complex gestures. Furthermore, the *OpenEarable* dashboard was augmented

[5]https://www.mixamo.com/
[6]https://www.blender.org
[7]https://modelviewer.dev/

Figure 4: Augmented OpenEarable Dashboard, showcasing the recognition of a left roll in the head gesture recognition module in the bottom right

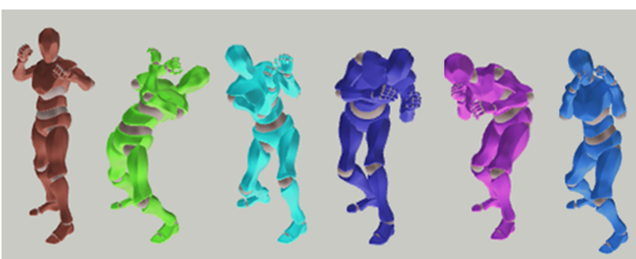

Figure 5: All model movements featured in the augmented dashboard, mid animation - From left to right: Idle, Left Slip, Right Slip, Left Roll, Right Roll, Pullback

to communicate with the server through a RESTful API. The developed system achieved a 96% accuracy in real-time, live testing, where recognized gestures were mirrored by a gesture mirror as feedback to the user.

Future research directions include further optimization of this system, such as employing a socket-based solution instead of relying on RESTful-based communication. Additionally, the system could be expanded upon by adding a feature wherein users can train and deploy any type of gesture directly from the *OpenEarable* user interface.

Acknowledgments

This study received partial funding from the ECOLOGIC project, which was (financially) supported by the Dutch Ministry of Infrastructure and Water Management and TKI Dinalog (case no. 31192090).

References

[1] William Cheung and Sudip Vhaduri. 2020. Context-Dependent Implicit Authentication for Wearable Device Users. In *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, London, United Kingdom, 1–7. https://doi.org/10.1109/PIMRC48278.2020.9217224

[2] Miguel Grinberg. 2018. *Flask web development: developing web applications with python.* " O'Reilly Media, Inc.".

[3] Alessandro Montanari, Andrea Ferlini, Ananta Narayanan Balaji, Cecilia Mascolo, and Fahim Kawsar. 2023. EarSet: A Multi-Modal Dataset for Studying the Impact of Head and Facial Movements on In-Ear PPG Signals. *Scientific Data* 10, 1 (Dec. 2023), 850. https://doi.org/10.1038/s41597-023-02762-3

[4] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12, Oct (2011), 2825–2830.

[5] François Petitjean, Germain Forestier, Geoffrey I Webb, Ann E Nicholson, Yanping Chen, and Eamonn Keogh. 2014. Dynamic time warping averaging of time series allows faster and more accurate classification. In *Data Mining (ICDM), 2014 IEEE International Conference on*. IEEE, 470–479.

[6] Tobias Röddiger, Christopher Clarke, Paula Breitling, Tim Schneegans, Haibin Zhao, Hans Gellersen, and Michael Beigl. 2022. Sensing with Earables: A Systematic Literature Review and Taxonomy of Phenomena. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 3 (Sept. 2022), 1–57. https://doi.org/10.1145/3550314

[7] Tobias Röddiger, Tobias King, Dylan Ray Roodt, Christopher Clarke, and Michael Beigl. 2022. OpenEarable: Open Hardware Earable Sensing Platform. In *Proceedings of the 2022 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, Cambridge United Kingdom, 246–251. https://doi.org/10.1145/3544793.3563415

[8] Stan Salvador and Philip Chan. 2007. FastDTW: Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis* 11, 5 (2007), 561–580.

[9] Dhruv Verma, Sejal Bhalla, Dhruv Sahnan, Jainendra Shukla, and Aman Parnami. 2021. ExpressEar: Sensing Fine-Grained Facial Expressions with Earables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 3 (Sept. 2021), 1–28. https://doi.org/10.1145/3478085