FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Real-Time Gesture Learning for Inclusive Music-Making: A Case Study with the Digiball

Alexandre Costa

Mestrado em Engenharia Informática e Computação

Supervisor: Prof. Gilberto Bernardes

Second Supervisor: Óscar Rodrigues

July 31, 2025

Real-Time Gesture Learning for Inclusive Music-Making: A Case Study with the Digiball

Alexandre Costa

Mestrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

President: Prof. Augusto de Sousa
Referee: Prof. Filipe Lopes
Referee: Prof. Gilberto Bernardes

July 31, 2025

Resumo

Esta dissertação apresenta o desenvolvimento, integração e avaliação de um sistema de reconhecimento de gestos acessível e expressivo para o Digiball, um instrumento musical digital universal desenvolvido pela Digitópia na Casa da Música. O principal objetivo do sistema é permitir que utilizadores, incluindo pessoas com dificuldades físicas ou cognitivas, possam definir mapeamentos personalizados entre gestos e som, sem necessidade de conhecimentos técnicos especializados.

Para responder às exigências de desempenho em tempo real e de reconhecimento com poucos dados, comuns em contextos participativos de criação musical, o sistema integra dois algoritmos de reconhecimento baseados em templates: o Dynamic Time Warping (DTW) e o Protractor3D. Foi desenvolvido um interface de treino no frontend e um backend em Python para suportar a gravação de gestos, a gestão de templates e o reconhecimento em tempo real a partir de dados de sensores inerciais.

A avaliação incluiu um estudo de campo com membros da associação Somos Nós, uma organização que trabalha com jovens adultos com dificuldades intelectuais e desenvolvimentais, bem como testes quantitativos em ambiente de laboratório. Os resultados evidenciam os compromissos entre os diferentes classificadores, nomeadamente no que diz respeito à sensibilidade ao tamanho do conjunto de treino e à qualidade da segmentação dos gestos. O estudo de campo demonstrou o potencial de envolvimento criativo dos utilizadores, mas também revelou desafios significativos relacionados com a fiabilidade do reconhecimento.

Este trabalho contribui para o avanço da investigação na área dos instrumentos musicais digitais acessíveis, demonstrando que algoritmos leves e baseados em templates podem ser implementados em sistemas adaptáveis e em tempo real, promovendo a inclusão e a expressividade na criação musical.

Abstract

This dissertation presents the design, implementation, and evaluation of an accessible and expressive gesture recognition system for the Digiball, a universal digital musical instrument developed by Digitópia at Casa da Música. The goal of the system is to enable users, including individuals with physical or cognitive disabilities, to define personalized gesture-to-sound mappings without requiring technical expertise.

To meet real-time and low-data constraints common in participatory music contexts, the system integrates two template-based gesture recognition algorithms: Dynamic Time Warping (DTW) and Protractor3D. A frontend training interface and a Python-based backend were developed to support user-driven gesture recording, template management, and real-time recognition using IMU data.

The system was evaluated using both qualitative field studies with members of Somos Nós, a community organization for young adults with intellectual and developmental difficulties, and controlled lab-based quantitative testing. Results highlight trade-offs between classifiers, especially in sensitivity to training set size and segmentation quality. The field study revealed user engagement potential but also exposed usability challenges related to recognition reliability.

This work contributes to ongoing research in accessible digital musical instruments by demonstrating how lightweight, template-based gesture recognition algorithms can be deployed in real-time, user-adaptive systems for inclusive music-making.

UN Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) provide a global framework for addressing urgent challenges such as inequality, innovation, and sustainable communities. This dissertation aligns with several of these goals, particularly those focused on inclusion, education, innovation, and reduced inequalities, namely SDG 4, SDG 9, and SDG 10.

The work presented in this dissertation contributes to SDG 4, Quality Education, by promoting inclusive and equitable access to music-making tools through the Digiball, a digital musical instrument designed with accessibility in mind. By enabling users with physical and cognitive disabilities to define their own gesture-to-sound mappings, the system fosters personalized learning and creativity, removing technical barriers to expressive participation in music.

It also supports SDG 9, Industry, Innovation, and Infrastructure, through the development of novel, lightweight gesture recognition algorithms that can function in real-time, with minimal training data. These technical innovations contribute to building adaptive digital infrastructures for creative expression, particularly in educational and community settings where traditional systems may be inaccessible or overly rigid.

Finally, the participatory methodology and real-world field study carried out with members of Somos Nós, an organization supporting young adults with intellectual and developmental disabilities, directly address SDG 10, Reduced Inequalities. By centering the needs and input of users often marginalized in digital musical interaction, the system aims to foster inclusion and reduce disparities in cultural and creative participation.

Table 1: SDGs' targets covered by this dissertation.

SDG	**Target**	**Contribution**	**Performance Indicators and Metrics**
4. Quality Education	4.5	Eliminates disparities in access to music education by supporting inclusive and personalized learning through accessible DMIs.	*Indicator 4.5.1*: Parity indices (female/male, rural/urban, disabilities, etc.) *Metric*: Increased access to participatory music-making for users with disabilities.
9. Industry, Innovation and Infrastructure	9.5	Fosters inclusive innovation in the creative industries through the development of lightweight, user-adaptive gesture recognition systems.	*Indicator 9.5.1*: Research and development expenditure *Metric*: Number of novel algorithms implemented and tested in inclusive contexts.
10. Reduced Inequalities	10.2	Empowers individuals with intellectual and developmental disabilities to participate equally in cultural and creative activities.	*Indicator 10.2.1*: Proportion of people living below 50% of median income, by age, sex and disability status *Metric*: Active participation of users with disabilities in system design and testing; qualitative feedback on creative agency.

Agradecimentos

I would like to thank my supervisor, Gilberto Bernardes, for his help in addressing technical challenges during the development of this dissertation. His input was especially valuable when solving specific implementation issues.

I am also very grateful to my supervisor, Óscar Rodrigues, for his continuous support and guidance throughout the entire process. His feedback and involvement were essential in shaping both the system and the dissertation itself.

I would like to thank Casa da Música for providing this opportunity, as well as for offering the space, resources, and materials that made this project possible.

I would also like to thank the team at Ligados às Máquinas for welcoming us and allowing valuable observations that helped inform the design process.

A sincere thank you as well to the participants from Somos Nós, whose engagement and collaboration during the evaluation phase were fundamental to testing the system in real-world conditions.

Finally, I would like to thank my girlfriend, Rita, for her emotional support and encouragement during this journey. Her patience and understanding helped me navigate the more challenging moments of this process.

Alexandre Costa

Contents

List of Figures

List of Tables

Abreviaturas e Símbolos

ACM	Association for Computing Machinery
ADMI	Accessible Digital Musical Instrument
AUMI	Adaptive Use Musical Instrument
CMMR	Computer Music Multidisciplinary Research
DCA	Data Combination Algorithm
DMI	Digital Musical Instrument
DTW	Dynamic Time Warping
FN	False Negative
FP	False Positive
HCI	Human-Computer Interaction
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertial Measurement Unit
IPSS	Instituição Particular de Solidariedade Social
IRCAM	Institut de Recherche et Coordination Acoustique/Musique
ISMM	IRCAM Sound Music Movement group
JSON	JavaScript Object Notation
k-NN	k-Nearest Neighbors
MOCO	Movement and Computing
NIME	New Interfaces for Musical Expression
PCA	Principal Component Analysis
RNBO	Real-time Neural-Based Object
SDGs	Sustainable Development Goals
SIGCHI	Special Interest Group on Computer–Human Interaction
SMC	Sound and Music Computing
TN	True Negative
TP	True Positive
UNESCO	United Nations Educational, Scientific and Cultural Organization

Chapter 1

Introduction

This dissertation presents the development, integration, and evaluation of a system designed to enhance the Digiball, an instrument developed by Digitópia and described as a "universal instrument." The system aims to improve the instrument's expressive potential through the use of traditional, non-deep-learning algorithms, specifically template matching and time-series distance metrics, enabling users to define personalized gestures for expressive musical interaction.

1.1 Context

Casa da Música is a landmark concert hall and cultural institution located in Porto, Portugal. It is widely known for its architectural significance and its role as a hub for musical education, performance, and cultural exchange. Casa da Música hosts various programs and initiatives to engage the local community and international audiences.

Digitópia is a department of Casa da Música that aims to support users in accessing digital environments focused on musical creation, thus promoting the exchange of ideas and experiences in a non-formal and informal teaching and learning model. With a focus on software design, music education, social inclusion, and collaborative musical creation, all based mostly but not primarily on digital tools, Digitópia seeks to create multicultural communities of musicians, composers, performers, and music enthusiasts.

Over the years, Digitópia has developed various educational software programs for concerts, workshops, and training sessions. After an initial phase focused on Mac OS and Windows operating systems, with examples such as Políssonos and Narrativas Sonoras, and a second phase aimed at mobile devices (iOS and Android), with examples like PortoPHONE and Gamult, recently, the development has shifted to being web-centered, due to its accessibility, with software such as 0+1=SOM and DigiCanvas.

Digitópia has also been developing new instruments referred to as *universal instruments*, tools designed so that anyone, including individuals with physical or cognitive disabilities, can engage with them. This concept aligns with principles of *universal design*, which aim to create products and environments usable by as many people as possible, without the need for specialized adaptation [17]. One such instrument is the Digiball, an instrument recently developed by Digitópia, which consists of a foam ball embedded with an Arduino microcontroller, equipped with a gyroscope and accelerometer. This setup captures motion data and transmits it to a web-based application that maps physical movement to sound. The foam material and spherical shape provide a soft, tactile, and playful interaction surface that distinguishes it from traditional digital interfaces.

1.2 Motivation

Although music is often described as a universal language, this idea is a misleading simplification that fails to account for the real barriers many people face in accessing musical tools and practices. While music exists across all societies and has a capacity for emotional and social connection, the means of making music are far from universally accessible. This romanticized view of universality can obscure the structural and design limitations that exclude many potential participants, particularly individuals with physical or cognitive disabilities.

Even when musical tools are made technically accessible, they often fall short in enabling expressive control. Everyone can make some form of sound, but not everyone can shape that sound with nuance, intention, or creative freedom. This distinction between basic access and expressive agency is central to the design concerns of this dissertation.

As ethnomusicologist John Blacking argues, all humans are innately musical, and music is a fundamental form of human expression present across all societies [5]. Similarly, UNESCO affirms music's role as a universal cultural resource that fosters dialogue and understanding [1]. However, the universality of music as a phenomenon does not imply universal access to the tools and environments required to engage with it. Instruments, interfaces, and educational systems are often designed with normative assumptions about users' physical and cognitive abilities. As such, individuals who deviate from these norms are frequently marginalized in musical contexts.

It is essential to distinguish between inclusion and participation. Inclusion often entails placing individuals into existing systems with minimal adaptation, expecting them to conform to environments not designed with their needs in mind. In contrast, participation requires systems to adapt to individuals, recognizing and responding to their diverse abilities, preferences, and forms of expression [3, 2]. As Peter Beresford emphasizes, meaningful participation involves power, choice, and agency.

This perspective is supported by Nussbaum's Capabilities Approach, which asserts that justice must be measured not merely by access or resources, but by what individuals can do and be [19]. In this view, technologies should expand people's capabilities in ways that enable them to engage meaningfully with cultural and creative practices, such as music making.

My involvement with Digitópia and my interest in music technology have made it clear how much room there is to expand these freedoms through inclusive design. Digitópia's approach, particularly its development of universal instruments like the Digiball, embodies a step toward such participatory engagement. However, much work remains to ensure these tools are accessible but also responsive and empowering. This dissertation addresses that gap by exploring how user-defined gesture learning can expand the creative and expressive possibilities available to all users, regardless of ability. Expanding possibilities involves two key goals: providing access to musical interaction and enabling users to shape their interaction methods. Together, these foster creativity, agency, and deeper engagement with music-making.

1.3 Problem

Despite growing interest in developing accessible digital musical instruments (ADMIs), many systems still fail to support the full range of physical and cognitive diversity among users [9].

Although developers have made progress in creating more inclusive hardware interfaces and musical environments [6], they often design software that rigidly governs user interaction, relying on fixed, predefined mappings between movement and sound output.

The Digiball exemplifies this limitation. Although its physical form is both inclusive and engaging, the interaction model currently offers the exact gesture-to-sound mapping to all users. Such uniform mappings constrain the instrument's versatility and reduce its adaptability for users with different motor profiles, including those with a limited range of motion or involuntary movement [14, 7]. Because the system does not adjust to individual differences in motion, users are limited in how effectively they can shape the qualities of the sound, such as timbre, texture, or dynamics. This lack of personalization reduces creative control and can restrict meaningful musical engagement.

As part of exploring alternative interaction designs that could improve adaptability, one option considered was replacing the Arduino with a smartphone as an alternative motion sensing device, since it includes built-in gyroscopes and accelerometers. However, this approach was rejected due to a lack of tactile, embodied qualities typically associated with expressive musical instruments. Prior research highlights how the physical form of an instrument, its weight, shape, material, and responsiveness, is critical in enabling expressive, sensorimotor interaction [4, 25]. Flat touch-screen devices, such as smartphones, are often optimized for visual interaction and do not provide the affordances needed for gestural nuance in musical performance [22].

One potential solution to improve accessibility could involve offering a library of predefined mappings tailored to various user profiles. While this might address some cases, the approach has limitations. The wide variation in users' motor abilities makes it difficult to anticipate every interaction style [10]. Moreover, static presets may not provide the nuanced control or expressive flexibility needed for creative exploration. Although defining or training gesture sound relationships may still involve navigating menus or setup interfaces, personalizing these mappings allows users to shape the instrument in ways that better reflect their intentions. This flexibility can ultimately reduce the need for ongoing adjustments and increase the system's long-term accessibility and usability [18].

1.4 Research Questions

This raises the central research question of this dissertation: *How can the Digiball be made more accessible and versatile through its software?* To address this, the proposed approach integrates traditional gesture learning algorithms into the Digiball's web-based application. Researchers have demonstrated that adaptive systems based on user-defined input profiles can respond to individual user behavior. This adaptability allows for flexible mappings, even when users generate noisy or non-standard input patterns [12, 8].

In this context, expressiveness refers to a user's ability to control and modify the qualities of the sound output in a way that reflects their intent. For example, a user could define a circular motion to gradually shift the sound from smooth to distorted, or a sharp shake to alter pitch or rhythm. Enabling users to define their gesture sound relationships empowers them to create personalized expressive vocabularies. The discussed concept aligns with the goals of participatory design, where systems evolve in response to those who use them [3].

By incorporating traditional algorithmic gesture learning methods into the Digiball's architecture, this dissertation aims to enhance its adaptability and expressiveness, supporting more inclusive, personalized, and creatively empowering forms of musical interaction.

Considering these challenges and goals, the following research questions guide this dissertation:

- How can traditional, non-deep-learning algorithms be used to learn user-defined gestures and enable personalized mapping between movement and sound?
- What design strategies can support users in creating their gesture sound relationships while maintaining ease of use and accessibility?
- How can the effectiveness of a gesture learning based system be evaluated in terms of its impact on expressiveness, user satisfaction, and inclusivity?

Addressing these research questions required a methodology capable of balancing technical implementation with iterative, user-centered design. The following section outlines the methodological approach adopted throughout this project.

1.5 Methodological Approach

The development of this dissertation followed an iterative and user-centered process, typical in human-computer interaction (HCI), music technology, and inclusive design projects. The primary objective was to design, implement, and evaluate a functional system that addressed the research questions while responding to real-world user needs and technical constraints.

The project was structured around the following iterative phases:

- **Initial Research and Requirement Gathering:** The process began with a review of existing literature on accessible digital musical instruments (ADMIs), gesture recognition methods, and inclusive interaction design. Informal conversations with the Digitópia team at

Casa da Música and early visits to Ligados às Máquinas provided valuable context regarding user needs, particularly in terms of physical accessibility and musical expressivity.

- **Design and Technical Implementation:** Based on the identified requirements and research gaps, a gesture recognition system was designed and implemented to integrate with the existing Digiball platform. Design decisions prioritized real-time performance, low-data training capability, and usability for non-technical users. The system architecture combined a Python-based backend with a web-based frontend, following participatory design principles.

- **Iterative Testing and Refinement:** Early versions of the system underwent informal testing with project stakeholders at Digitópia. These iterative cycles allowed for continuous refinement of both the user interface and the gesture recognition backend. Feedback during this phase focused on improving system responsiveness, usability, and technical robustness.

- **Formal Evaluation:** The final system was evaluated using two complementary methods: (i) a *field study with users from Somos Nós*, focusing on real-world usability, accessibility, and user experience; and (ii) a *controlled lab-based evaluation*, which assessed quantitative performance metrics such as precision, recall, and accuracy.

  Both evaluation phases informed the final discussion and conclusions, providing critical insights into system performance and user engagement.

Throughout the project, the development process remained iterative, with system adjustments made primarily in response to internal testing and the results observed during the first field evaluation. Although time constraints limited the extent of user-driven refinements, the insights gathered after the initial session informed targeted improvements before the second round of testing.

1.6 Design Practice as a Methodological Framework

The methodological approach adopted in this dissertation aligns with the paradigm of *design practice*, where design is not only a means to solve predefined problems but also a process of inquiry in its own right [28, 11]. This framework treats the act of designing as a form of research that generates knowledge through iterative cycles of creation, testing, and reflection in real-world contexts.

Rather than validating hypotheses through controlled experimentation alone, *design practice* investigates possible futures by prototyping functional systems that respond to the needs of specific user groups. In this case, the work focused on enabling expressive musical interaction for individuals with physical and cognitive disabilities, a context that demands participatory, adaptable, and situated design methods.

This project followed a cyclical process of ideation, implementation, and evaluation. Insights from preliminary observations and the first field session led directly to system refinements that were tested in the second session. These cycles exemplify *design practice's* emphasis on iterative refinement and responsiveness to context. Moreover, the evaluation itself was not merely summative but served as an integral part of the design process, a source of insight and inspiration for further development.

Chapter 2

Literature Review and State of the Art

Developing digital musical instruments (DMIs) that are both expressive and accessible requires engagement with multiple research fields, including music technology, human-computer interaction (HCI), disability studies, and gesture recognition. This chapter presents a review of the relevant literature, mapping key themes and identifying research gaps that informed the design decisions in this dissertation.

Rather than conducting a systematic literature review, this chapter adopts a scoping approach, progressively narrowing its focus from broad themes like music and accessibility to gesture recognition and, finally, gesture recognition in music contexts.

2.1 Scoping Review Process and Overview

The scoping review conducted for this dissertation aimed to capture key research themes across gesture recognition, accessible DMIs, and inclusive design in music technology. The literature search covered publications from 2010 to 2025 and included peer-reviewed journal articles, conference proceedings, and relevant technical reports.

Databases searched included Google Scholar, IEEE Xplore, Web of Science, and the ACM Digital Library. Key conferences and venues consulted were the International Conference on New Interfaces for Musical Expression (NIME), the International Symposium on Computer Music Multidisciplinary Research (CMMR), and the International Symposium on Movement and Computing (MOCO) [15].

Inclusion criteria prioritized works addressing gesture recognition, musical interaction, accessibility, and participatory design. Exclusion criteria ruled out papers unrelated to real-time interaction, music performance, or user-driven system customization. This multi-source review process ensured a comprehensive understanding of the state of the art in both technical and human-centered aspects of the field.

2.2 Music and Expressivity

Several digital musical instruments (DMIs) have been developed to support creative expressivity, particularly for users with diverse physical abilities. For example, the Soundbeam system uses ultrasonic sensors to translate movement into sound, making it popular in special education settings. The Adaptive Use Musical Instrument (AUMI) allows users to control musical output using small, intentional gestures captured by a webcam [23]. Another example is the EyeHarp, a gaze-controlled DMI that enables users with severe motor disabilities to play melodies using only their eye movements [27].

These projects illustrate the balance between accessibility and expressivity that underpins much DMI research. While many systems enable basic sound triggering, fewer offer the nuanced control needed for expressive performance [9]. This dissertation builds upon these lessons by focusing on customizable gesture-to-sound mappings that support both access and creative agency [18].

2.3 Disability and Inclusive Design

Inclusive design in music technology focuses on creating tools that are usable by individuals with a wide range of physical and cognitive abilities. The Universal Design framework promotes accessibility without the need for later adaptation, but achieving this in music technology remains challenging [20].

Participatory design approaches, such as those used in the AUMI project and Digitópia's universal instruments, emphasize direct involvement of target users in system development [9]. Other projects like the MiMu Gloves provide gestural control for users with limited dexterity, though such systems often remain financially inaccessible [13].

Despite these advances, many existing tools prioritize basic access at the expense of expressive nuance. Moreover, few tools offer end-users the ability to configure or train their own gesture vocabularies without technical assistance. This gap in accessible and customizable interaction design motivates the system described in this dissertation.

2.4 Gesture Recognition

Gesture recognition refers to the automated detection and classification of human movement patterns, often using data from sensors like accelerometers and gyroscopes [16]. Techniques vary widely in terms of data requirements, computational cost, and suitability for real-time user-driven interaction.

2.4.1 Gesture Recognition Algorithms

Gesture recognition algorithms typically fall into three broad categories: template-based methods, classical machine learning classifiers, and deep learning models.

Template-based methods, such as Dynamic Time Warping (DTW) and Protractor3D, require small amounts of training data and are computationally lightweight [21, 16]. They are well-suited for real-time interaction but may struggle with high intra-class variability.

Classical machine learning models, including Support Vector Machines (SVMs) and k-Nearest Neighbors (k-NN), offer better generalization but require larger, cleaner training datasets and more computational resources.

Deep learning models, while state-of-the-art in many gesture recognition benchmarks, demand extensive labeled datasets, high computational power, and are generally unsuitable for low-latency, user-configurable musical contexts [12].

Table 2.1 summarizes the trade-offs between these approaches.

Table 2.1: Comparison of Gesture Recognition Approaches

Method	Data Requirement	Latency	Flexibility	Real-Time Suitability
DTW	Low	Low	Medium	High
Protractor3D	Low	Very Low	Low-Medium	High
SVM / k-NN	Medium	Medium	Medium	Medium
Deep Learning	High	High	High	Low

2.4.2 User-Defined Gesture Mapping

Enabling user-defined gesture sets is a key challenge in gesture recognition for DMIs. Most systems assume fixed gesture vocabularies or require extensive offline training phases, creating barriers for non-technical users [8]. Few-shot learning approaches, like DTW and Protractor3D, offer a pathway to overcome these barriers, allowing systems to generalize from small training sets while maintaining real-time performance [21, 16].

2.4.3 Real-Time Interaction Constraints

Musical applications impose strict real-time performance constraints on gesture recognition systems. Latency and computational load directly affect a performer's ability to engage expressively [12]. Lightweight algorithms like template matching provide better real-time performance, but often at the cost of reduced classification precision.

2.4.4 Gesture Recognition in Music

Gesture recognition in musical contexts presents unique challenges compared to other domains like gaming or robotics. Real-time responsiveness, low-latency performance, and user-configurable gesture mappings are critical for expressive interaction [25].

Tools like the Wekinator enable interactive machine learning for musical gesture mapping but require substantial training data and technical expertise, making them less accessible for participatory or community-based contexts [8]. Furthermore, most published gesture recognition systems have not been tested with users with disabilities, creating a significant gap in knowledge about how such systems perform under diverse motor conditions [9].

This dissertation addresses these gaps by exploring template-based gesture recognition approaches that offer low-data, real-time classification suitable for inclusive musical performance contexts.

2.5 Identified Gaps and Challenges

The reviewed literature highlights several critical gaps and challenges that this dissertation seeks to address:

- **Support for Non-Normative Gestural Input:** Existing systems are typically trained on data from non-disabled users, with little research on how recognition models handle irregular or noisy gestures common among users with motor impairments [9].
- **User-Driven Gesture Training and Mapping:** Few systems provide intuitive interfaces that allow non-technical users to train and configure their own gesture vocabularies in real-time [18].
- **Balancing Real-Time Performance and Adaptability:** Many high-accuracy models sacrifice latency, making them unsuitable for live musical use. Lightweight models often trade off classification precision for speed [12].
- **Low-Data Learning Strategies:** There is a need for recognition methods that perform reliably with small training datasets, suitable for end-user-driven customization [16].
- **Inclusive Evaluation Frameworks:** Most evaluation studies focus on technical performance metrics like accuracy, neglecting measures related to user experience, perceived expressivity, and inclusivity.
- **Bridging Access and Expression:** Current tools often provide basic access but lack mechanisms for nuanced, expressive control over sound output.

These gaps point to a pressing need for gesture recognition systems that can accommodate diverse users and contexts without sacrificing responsiveness or creative potential. Many existing ADMIs either lack the flexibility for user-driven customization or fall short in supporting nuanced, expressive control, particularly in real-time scenarios involving users with motor variability. Bridging this divide requires a more critical look at recent developments in the field. While several innovative ADMIs have emerged, they often highlight the trade-offs between accessibility and expressivity. The Adaptive Use Musical Instrument (AUMI), for instance, allows users with minimal mobility to trigger sound via webcam input, but generally supports only basic sonic output with limited expressive control. The EyeHarp, which translates eye gaze into melodic input, offers more sophisticated musical possibilities but requires visual focus and postural stability that may exclude users with complex motor challenges. On the other end of the spectrum, the MiMu Gloves, worn by artists like Imogen Heap, enable highly expressive gestural performance, but their cost, complexity, and reliance on external software tools make them less accessible in educational or community-based settings.

These examples reinforce the need for mid-spectrum solutions: systems that are expressive enough to empower users creatively, yet simple and adaptable enough to be inclusive. The gesture recognition system proposed in this dissertation aims to fill that gap by using lightweight, customizable algorithms suited for non-technical users in participatory music contexts. To support this analysis, illustrative images of three accessible musical interfaces are provided. Figure 2.1a shows the interface of the EyeHarp app, which enables melodic control via gaze tracking. Figure 2.1b depicts the AUMI iOS application interface, designed to trigger sounds through small intentional movements captured by a webcam. Figure 2.1c presents the MiMu Gloves, a wearable gestural interface used in professional music performance. While these figures do not show real-world usage scenarios, they visually highlight the differing interaction models, levels of technical complexity, and aesthetic design choices behind each system. These contrasts help contextualize the system developed in this dissertation as a low-cost, adaptable solution focused on personalized gesture input and inclusive creative expression.

(a) EyeHarp app

(b) AUMI iOS app

(c) MiMu gloves

Chapter 3

Implementation: Extending the Digiball System

This chapter describes the system architecture and implementation details of the developed gesture recognition and sound mapping platform. The system was designed to enable users to define, train, and classify personalized gestures in real-time, triggering sound responses with minimal technical barriers.

3.1 System Overview

The system follows a modular, frontend-driven architecture composed of three main layers: data acquisition, gesture recognition, and sound generation (Figure 3.1).

The first layer captures motion data from an inertial measurement unit (IMU) embedded in the Digiball device. The second layer processes and classifies the captured data through a backend service. Finally, the third layer handles the sound generation, providing real-time auditory feedback to the user.

3.2 System Architecture

Figure 3.1 presents an overview of the system architecture and data flow.

The system starts with the IMU sensor embedded in the Digiball device, which streams motion data directly to the frontend web application developed in *p5.js*. The frontend continuously processes incoming data and determines behavior based on the active user interface state:

- **Training Mode:** When the user is on the training page, the frontend collects labeled gesture samples and sends them to the backend for storage as templates.
- **Playing Mode:** When the user is on the playing page, the frontend monitors for motion events. Upon detecting movement, it segments the relevant IMU data and sends it to the backend for classification.

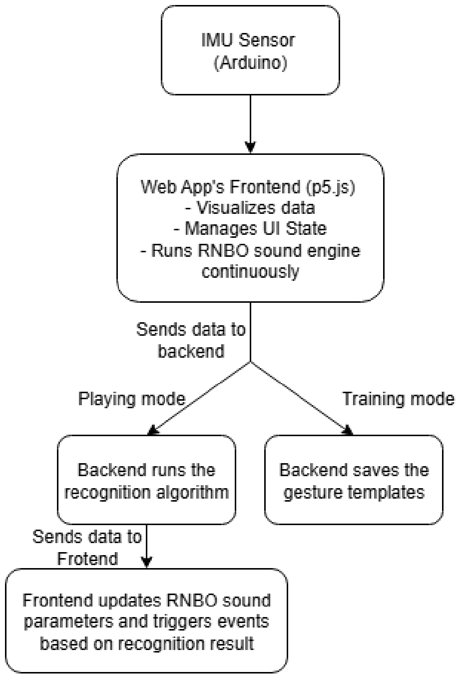

Figure 3.1: System architecture and data flow showing the interaction between the IMU sensor, frontend web application, backend services, and RNBO sound engine.

The backend, implemented in Python, either stores gesture templates or performs classification using DTW or Protractor3D. Classification results are returned to the frontend, which uses them to trigger corresponding sound responses within the RNBO sound engine.

3.2.1 Component-Level Breakdown

While Figure 3.1 presents a high-level view of data flow between the IMU, frontend, backend, and RNBO engine, this section expands on the internal responsibilities and processes handled by each component. These details help clarify the role and structure of each module in supporting real-time, user-configurable gesture recognition.

3.2.1.1 IMU Sensor and Data Transmission

The Digiball device contains an IMU Shield connected to an Arduino microcontroller, which captures 3-axis accelerometer and 3-axis gyroscope data. This motion data is transmitted wirelessly over Bluetooth Low Energy (BLE) to the host machine. A Python-based bridge receives the BLE packets and re-publishes them to the browser frontend over WebSockets. The sensor data is relayed with minimal delay and without pre-filtering, preserving temporal structure for later segmentation and classification.

3.2.1.2 Frontend (p5.js) Application

The web frontend, developed in `p5.js`, serves as the main user interface and gesture processing layer. It supports two UI states, Training and Playing, which determine how incoming sensor data is handled. In Training mode, gestures are recorded on user command, labeled, and sent to the backend for template storage. In Playing mode, incoming data is continuously monitored using a variance-based segmentation algorithm that detects gesture onsets. This data is buffered in memory and, once a gesture is detected, the relevant segment is sent for classification. Data exchange between the frontend and backend is handled asynchronously via WebSockets, using JSON messages.

3.2.1.3 Backend Recognition Pipeline

The backend is written in Python and handles template storage, gesture classification, and communication. Its key components include:

- **Template Storage:** Labeled gesture templates are stored in memory as sequences of 6D sensor vectors, allowing future comparison during classification.
- **Segmentation Validator:** Before classification, gesture segments are validated to ensure they fall within acceptable length and motion intensity bounds.
- **Classifier Engine:** The backend supports both Dynamic Time Warping (DTW) and Protractor3D for gesture recognition. While there is no dynamic switch between the two, different sessions can be manually configured to use one or the other. Both algorithms compare new gestures against stored templates and return the label of the closest match (or a null result if below threshold).

3.2.1.4 RNBO Sound Engine Integration

The RNBO engine runs inside the frontend browser context and remains active during both training and interaction phases. Recognized gesture labels are mapped to combinations of:

- Triggered sound events (e.g., playing a sample or note),
- Selection of musical scales,
- Modulation of granular synthesis parameters (such as grain position or density).

These mappings are defined in code and designed to allow expressive and immediate sound feedback in response to user-defined gestures.

3.3 Data Flow and Communication Pipeline

The communication pipeline is designed for low-latency, real-time interaction and involves the following stages:

1. **IMU Sensor:** Streams raw accelerometer and gyroscope data to the frontend.
2. **Frontend Web Application (p5.js):**
   - Continuously manages the UI state (Training or Playing mode).
   - Runs the RNBO sound engine continuously.
   - Sends labeled gesture samples to the backend for storage (during training).
   - Sends segmented motion data to the backend for classification (during playing).
   - Updates RNBO sound parameters or triggers events based on classification results.
3. **Backend (Python):**
   - Stores gesture templates for later classification.
   - Runs recognition algorithms (DTW or Protractor3D) when classification requests are received.
   - Returns classification results to the frontend.
4. **RNBO Sound Engine (Frontend):** Continuously generates sound textures and responds to classification events.

3.4 Frontend Implementation

The frontend, built using *p5.js*, serves as the user-facing component of the system. It provides:

- Interfaces for recording and labeling gesture templates.
- Real-time monitoring and segmentation of motion during playing sessions.
- Communication with both the backend (for storage and classification) and the RNBO sound engine (for sound generation).

3.5 Backend and Recognition Pipeline

The backend was developed in Python and handles data storage and gesture recognition. Its main responsibilities include:

- **Template Storage:** Receives labeled gesture templates from the frontend and stores them for future classification.
- **Segmentation Validation:** Verifies that received segments are within the expected duration and quality for classification.
- **Classification:** Runs either DTW or Protractor3D algorithms based on the active configuration. Returns classification results (gesture label or no match) to the frontend.

During the design of the recognition system, existing research on IMU-based gesture classification helped guide the selection of gesture types and informed the understanding of motion shape variability relevant for this kind of recognition task. Figure 3.2 presents gesture shape examples provided by Kratz and Rohs (2013) [16], which served as conceptual references during development.

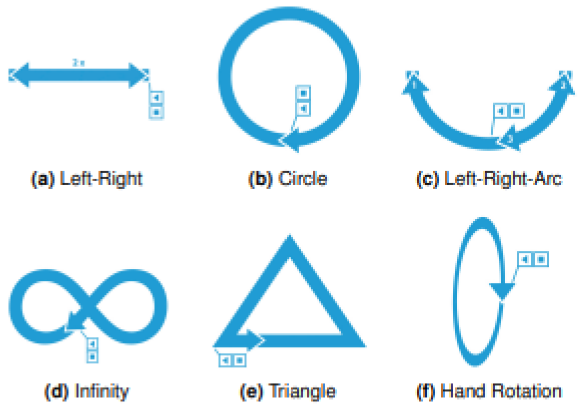

Figure 3.2: Examples of gesture shapes as described in Kratz and Rohs (2013) [16]. These served as conceptual references during the design and implementation of the recognition system.

The backend communicates with the frontend using a WebSocket-based protocol to ensure low-latency data exchange.

3.6 Sound Engine and RNBO Integration

The RNBO sound engine runs within the frontend web application and remains active during both training and playing modes. It handles the generation of continuous textures as well as discrete sound events triggered by classification results.

Upon receiving a recognized gesture label from the backend, the frontend modifies RNBO parameters or triggers specific sound events, enabling an immediate and responsive auditory experience for the user.

Chapter 4

Evaluation Methodology

This chapter outlines the methods used to evaluate the performance, usability, and accessibility of the gesture recognition system implemented for the Digiball. Two complementary evaluation approaches were conducted: a field study with participants from Somos Nós, and a controlled lab-based quantitative test measuring classification accuracy under varying training conditions.

4.1 Overview of Evaluation Approach

The evaluation strategy combined qualitative and quantitative methods to understand both user experience and technical system performance. The field study focused on user-centered outcomes such as usability, engagement, and accessibility, while the lab-based evaluation examined classifier performance under controlled conditions.

This mixed-methods design aligns with the system's dual objectives: enabling inclusive user interaction while ensuring reliable real-time classification with low-data training scenarios.

4.2 Field Study at Somos Nós

4.2.1 Context and Participants

Somos Nós is a non-profit organization (IPSS) dedicated to supporting young adults with intellectual and developmental difficulties. The institution promotes autonomy, self-esteem, and community participation through creative and educational activities.

Participants for the field study were regular members of Somos Nós, recruited based on willingness to participate and ability to interact with the Digiball with minimal physical assistance. No diagnostic labels were recorded to preserve privacy and inclusivity.

4.2.2 Session Planning and Procedure

The field evaluation was structured into two sessions, held one week apart. The main goals were to assess usability, system responsiveness, and user engagement with the gesture-sound mapping process. Each session lasted approximately 25 minutes and involved 2 to 3 participants. Two Digitópia team members were present to assist with gesture recording and configuring sound mappings. The setup included a laptop running the Digiball web application, connected via cable to a small amplifier. Sessions were held in a multi-purpose room provided by the Somos Nós organization.

Each session followed a three-phase structure:

- **Phase 1: Free Play** Participants explored the Digiball without guidance to observe natural interaction styles and baseline behavior.
- **Phase 2: Gesture Training** Participants were invited to define gestures and associate them with sound changes. In Session 1, participants freely invented gestures; in Session 2, they were asked to reproduce predefined gesture shapes (e.g., circles, lines) to reduce variability and improve classifier performance.
- **Phase 3: Testing and Interaction** Participants tested the system's ability to recognize the trained gestures and produce the expected sounds.

Between sessions, the system was refined based on observations from Session 1, particularly addressing classifier sensitivity to gesture variability and recognition errors.

4.2.3 Data Collection Methods

Qualitative data were collected through:

- **Observational Notes** Taken by the facilitator and support staff, focusing on user reactions, engagement levels, and ease of interaction.
- **Video Recordings** Used to document gesture execution, user-system interactions, and immediate behavioral responses.

No personally identifiable information was stored. All data collection followed ethical guidelines for working with vulnerable user groups.

4.2.4 Evaluation Expectations

Before the sessions, several hypotheses guided the evaluation:

- Participants would engage creatively with the gesture training interface.
- The system would successfully recognize simple, user-defined gestures with a limited number of templates.
- Recognition accuracy would likely be lower in Session 1 (free-form gestures) compared to Session 2 (standardized gestures).

- Latency and segmentation errors could introduce recognition failures.

These expectations reflected both insights from related work (as outlined in Chapter 2) and technical limitations identified during system development.

4.3 Lab-Based Quantitative Testing

4.3.1 Test Design

To complement the qualitative field study, a lab-based evaluation was conducted to measure classifier performance under controlled conditions.

Two distinct gestures were selected for testing, with three training set sizes per gesture: 5, 10, and 20 templates. This design tested the system's few-shot learning capability, reflecting project priorities on low-data adaptability.

For each training condition, 40 test trials were conducted:

- 20 gesture trials (10 per gesture).
- 20 noise trials (non-gesture movement segments).

Both manual and automatic segmentation modes were used, providing insight into the impact of segmentation quality on recognition performance.

4.3.2 Evaluation Metrics

Classifier performance was evaluated using the following metrics:

- **True Positives (TP)**
- **True Negatives (TN)**
- **False Positives (FP)**
- **False Negatives (FN)**
- Derived metrics: **Precision**, **Recall**, and **F1-Score**

Each classifier (DTW and Protractor3D) was tested under all segmentation and template-size conditions.

4.3.3 Lab Evaluation Expectations

Drawing from previous research and initial informal tests:

- DTW was expected to outperform Protractor3D at smaller training set sizes, given its time-series alignment strengths.

- Protractor3D was expected to show improved performance as the number of templates increased, due to its distance-based matching.
- Manual segmentation was anticipated to yield higher accuracy than automatic segmentation, given clearer gesture boundaries.

Unexpected behaviors and performance trade-offs observed during testing are discussed in detail in Chapter 5.

Chapter 5

Results and Discussion

This chapter presents and analyzes the results from the field study conducted at Somos Nós and the lab-based quantitative testing. It highlights key observations and system performance metrics, and reflects on how well the implementation addressed the research questions and challenges outlined in earlier chapters.

5.1 Field Study Results

5.1.1 Observations from the First Session (DTW with 3 Templates)

During the first evaluation session, where the DTW algorithm was used with three templates per gesture, several usability patterns emerged:

- **Gesture Complexity:** Participants often created long, complex, or highly variable gestures. This diverged significantly from the simple geometric gesture shapes originally envisioned during system design (e.g., circles, lines, triangles). This increased variability directly contributed to recognition inconsistencies. The system, trained on limited templates with relatively uniform characteristics, struggled to generalize across the wide range of participant-created gestures. This observation aligns with the challenges described in Chapter 2 concerning non-normative gesture input [10].

- **Assisted Recording and Personalization:** Gesture recording and sound mapping were completed with facilitator assistance to ensure accurate template capture and appropriate sound personalization. The sound customization options included scale changes, granular synthesis control, and sound triggering. This supported the project's enabling expressive, user-driven mappings, as discussed in Chapter 3.

- **Recognition Performance:** The system exhibited false positives and negatives. False positives occurred when unrelated movements triggered gesture recognition, while false negatives arose due to the high variability between training samples and test gestures. This behavior reflects the known trade-offs of template-based classifiers [16].

- **User Response:** Participants occasionally appeared confused or frustrated when the system failed to respond as expected, particularly during recognition. This suggests that recognition inconsistency can directly undermine user engagement and confidence [10].

5.1.2 Observations from the Second Session (Protractor3D with 5 Templates)

For the second session, Protractor3D was used with five templates per gesture. The following adjustments were made based on observations from the first session:

- **Gesture Standardization:** Participants were asked to perform predefined gesture shapes (e.g., circles and lines) to reduce variability and improve recognition accuracy. This addressed the unpredictability issues in the first session [26].
- **Recognition Performance:** No false positives were observed, demonstrating improved specificity. However, the system showed a higher rate of false negatives, leading to many missed recognitions. This trade-off between reducing false positives and increasing false negatives mirrors challenges discussed in Chapters 2 and 3.
- **User Response:** Despite more straightforward gesture guidelines, participants appeared even more troubled due to the increased number of failed recognitions. This highlights the tension between classifier specificity and sensitivity [8].

These field observations highlight the system's sensitivity to gesture variability and reinforce the need for future solutions to maintain performance under diverse input conditions.

5.2 Lab-Based Quantitative Testing Results

This section presents a detailed quantitative evaluation of the two gesture recognition classifiers: Protractor3D and DTW. The results discussed here are derived exclusively from the lab-based testing described in Section 4.3, where controlled experiments were conducted to assess classifier performance under varying template sizes and testing modes (manual and automatic). The metrics evaluated include Precision, Recall, F1-score, and Accuracy.

To assess classification performance, four standard evaluation metrics were used:

- Precision: The proportion of recognized gestures that were correct, reflecting the system's ability to avoid false positives.
- Recall: The proportion of actual performed gestures that were successfully recognized, measuring the system's ability to avoid false negatives.
- F1-Score: The harmonic mean of Precision and Recall, providing a balanced metric that considers both types of recognition errors.

- Accuracy: The overall proportion of correct classifications (both true positives and true negatives) out of all recognition attempts.

Table 5.1: Evaluation Metric Definitions

Metric	**Formula**
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F1-Score	$2 \cdot \frac{Precision \cdot Recall}{Precision+Recall}$
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$

Classifier performance was evaluated under two segmentation modes:

- Manual segmentation: Gesture recording was manually controlled, with start and stop points explicitly defined for each recognition attempt to ensure precise segmentation.
- Automatic segmentation: Gesture boundaries were detected using the system's built-in motion onset detection algorithm (described in Chapter 3, Section 3.2.2), which applies real-time variance-based thresholding over accelerometer and gyroscope data streams to identify the start and end of gesture events.

This comparison between manual and automatic segmentation provides insight into how the segmentation method impacts recognition performance.

5.2.1 Precision

Figure 5.1 shows how precision varied with training set size. DTW consistently outperformed Protractor3D at 5 and 10 templates, especially in automatic mode, which peaked around 0.69. However, at 20 templates, DTW-manual precision declined sharply, dropping below 0.4. Protractor3D, starting with lower precision, showed steady improvement with increasing template size, reaching over 0.55 in manual mode at 20 templates. This suggests that DTW is better at reducing false positives with fewer templates but struggles when the template pool grows too large and variable.

5.2.2 Recall

Figure 5.2 highlights recall trends. Protractor3D consistently increased recall with more templates, reaching perfect recall (1.0) in manual mode at 20 templates. While performing well at 10 templates, DTW showed a noticeable recall drop at 20 templates, particularly in manual mode. This suggests that Protractor3D is more robust in avoiding false negatives as training data increases, whereas DTW's recall diminishes under higher template variability.

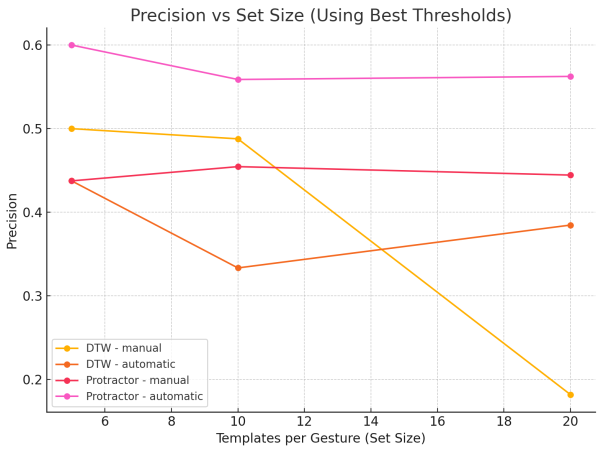

Figure 5.1: Precision at Different Training Sizes

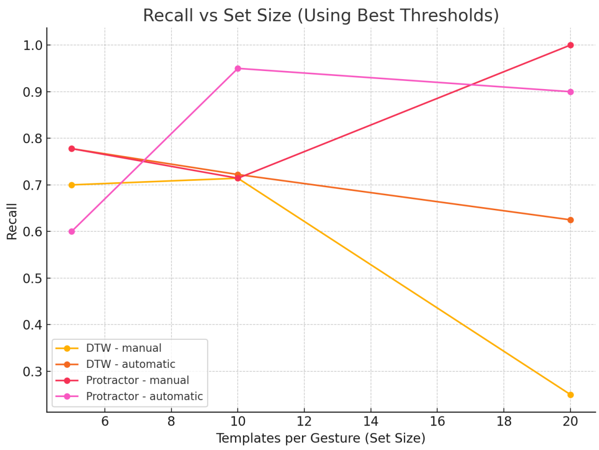

Figure 5.2: Recall at Different Training Sizes

5.2.3 F1-Score

The F1-score, shown in Figure 5.3, balances precision and recall. DTW-automatic achieved the highest F1-score (0.78) at 10 templates, marking an optimal balance under this condition. However, F1 dropped significantly in DTW-manual at 20 templates. Protractor3D, while initially lower, improved consistently with more templates, reaching approximately 0.70 in manual mode and slightly less in automatic mode. These results indicate that Protractor3D becomes more reliable as gesture data expands.

5.2.4 Accuracy

Figure 5.4 summarizes accuracy trends. DTW-automatic peaked at 10 templates with an accuracy of 0.75, the highest single-condition accuracy observed. However, the DTW manual fell sharply at 20 templates, aligning with its precision and recall drops. Protractor3D showed a smoother upward

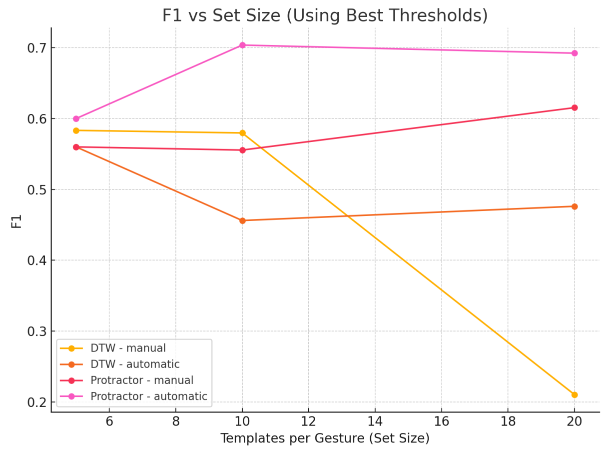

Figure 5.3: F1-Score at Different Training Sizes

trend, achieving around 0.60 accuracy in both modes at 20 templates. These findings reinforce the idea that DTW performs well with smaller, clean template sets, while Protractor3D offers greater stability at scale.

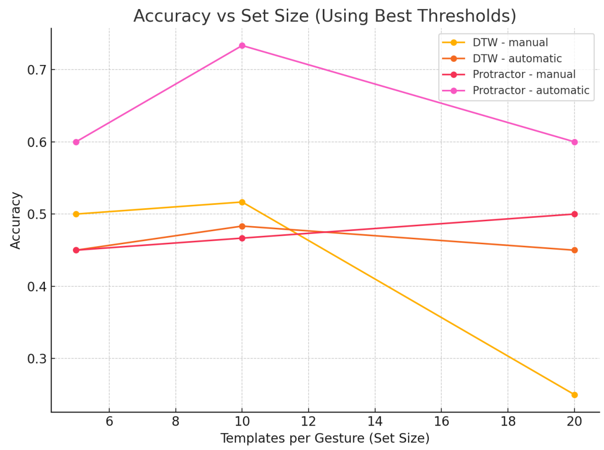

Figure 5.4: Accuracy at Different Training Sizes

5.2.5 Limitations

The quantitative results are based on a relatively small and specific sample. Statistical testing (e.g., significance or confidence intervals) was not performed. Additionally, segmentation inconsistencies and participant variability may have influenced classifier performance. Future work should consider controlled studies with larger sample sizes and statistical validation.

5.3 Discussion

5.3.1 Classifier Trade-offs

DTW performed best at lower template sizes, particularly at 10 templates, achieving high precision, recall, and F1 in automatic mode. However, its performance degraded sharply at 20 templates, especially in manual segmentation. This suggests a sensitivity to template variability or overfitting.

In contrast, Protractor3D exhibited consistent scalability. Its performance improved steadily with more templates, achieving competitive recall, F1, and accuracy at 20 templates. Protractor3D thus appears more robust when scaling gesture libraries or supporting diverse user input.

One notable observation from the quantitative tests was the decline in DTW performance when the number of training templates increased to 20. A likely contributing factor is the increased intra-class variability introduced by a larger and more diverse set of gesture examples. As the template pool grows, the classifier must compare each incoming gesture against a broader and less consistent set of samples, increasing the risk of poor matches and misclassifications. This sensitivity to template variability is a well-documented challenge in template-based recognition systems, particularly when training data is user-generated and collected under non-controlled conditions [16, 8, 12]. In such scenarios, differences in gesture execution style, speed, and amplitude between templates can reduce classification reliability, especially for algorithms like DTW that rely on distance minimization across all stored samples. Given that real-world deployments, such as community workshops or participatory music settings, are likely to involve even greater user variability, addressing this issue through improved normalization, template filtering, or the use of more variability-tolerant classifiers represents an important direction for future work.

5.3.1.1 Segmentation Mode Effects

Classifier performance varied by segmentation mode. DTW consistently performed better in automatic mode at higher template sizes, possibly due to more consistent gesture framing. Manual mode favored DTW only at smaller template sizes. Protractor3D showed less sensitivity to segmentation mode, suggesting better generalizability across modes.

Participant feedback also aligned with these trends: while automatic mode sometimes introduced confusion due to mis-segmented gestures, manual mode increased user control but added cognitive demand.

5.3.2 Overall System Performance and User Experience

Despite metric improvements, user experience during field studies highlighted the emotional impact of recognition failures (false positives and negatives), though quantifiable, translated to frustration and disengagement. Nonetheless, the system achieved its primary aim of enabling personalized gesture-sound mappings without requiring technical expertise. Protractor3D's performance at higher template sizes supports its suitability for community-oriented, user-led use cases.

5.3.3 Conclusions

The comparative evaluation of DTW and Protractor3D highlights necessary trade-offs for accessible, user-driven gesture recognition in digital musical instruments. DTW demonstrates superior precision with small, consistent training sets and may be better suited to facilitator-led sessions or highly constrained environments. However, its performance degrades with increasing template variability, limiting its use in participatory or exploratory contexts.

Protractor3D, by contrast, offers greater robustness under diverse user input conditions. Its accuracy and recall scale are good with template size, making it a stronger candidate for unsupervised or community-based deployments where users create gesture vocabularies. The absence of false positives in field testing reinforces its reliability in noisy, real-world settings, even though sensitivity remains a limitation.

Critically, the system met its goal of enabling user-defined, expressive interaction without requiring technical expertise. While gesture recognition errors sometimes led to user frustration, participants successfully configured personal mappings and engaged creatively with the instrument. These findings suggest that technical performance, while necessary, is insufficient, meaning that perceived responsiveness and reliability are equally crucial for sustained user engagement.

Future work should improve segmentation reliability, refine gesture training interfaces, and integrate real-time feedback to support learnability. A larger-scale deployment with a broader participant base and formal evaluation of expressive outcomes would further validate the system's potential as an inclusive musical interface.

Chapter 6

Conclusions and Future Perspectives

This research aimed to explore the integration of traditional gesture recognition algorithms, based on template matching techniques, into the Digiball system to enhance its expressiveness, adaptability, and inclusivity. The core challenge addressed was creating accessible digital musical instruments that enable users with diverse motor profiles to configure and control sound mappings intuitively, without requiring extensive technical expertise.

Throughout the development process, it was observed that lightweight, user-configurable classifiers such as Protractor3D offer a scalable solution for participatory and inclusive musical interaction contexts. However, these classifiers exhibit notable limitations regarding sensitivity to gesture variability and recognition accuracy. The field study and laboratory evaluations revealed that while the system facilitates user engagement and participatory music making, the high variability in user-generated gestures poses challenges to reliable recognition, often resulting in false positives and negatives.

6.1 Addressing the Research Questions

With the system implemented and evaluated through both participatory and technical methods, this section revisits the core research questions to reflect on how effectively the project met its objectives and contributed to inclusive musical interaction.

6.1.1 How can user-defined gestures be learned and used to enable personalized mapping between movement and sound?

User-defined gestures can be learned through systems that capture motion data, store labeled examples, and classify new input by comparing it to these templates. This enables personalized mapping by allowing users to define their own movements and associate them with specific sounds, creating a direct and custom interaction model.

In this project, gesture learning was achieved using two lightweight, template-based algorithms: Dynamic Time Warping (DTW) and Protractor3D. These algorithms were integrated into the Digiball's software pipeline, enabling gesture recognition with a small number of training examples. During the field study, participants were able to record their own gestures, either freely invented or based on predefined shapes, and use them to trigger sound events. Lab-based quantitative tests confirmed the system's ability to recognize gestures in real time, with results showing that DTW performed better at lower template counts, while Protractor3D scaled more reliably as the number of templates increased.

6.1.2 What design strategies can support users in creating their gesture-sound relationships while maintaining ease of use and accessibility?

Effective design strategies include simplifying the interface, reducing technical barriers, clearly separating training and interaction modes, and automating complex processes such as gesture segmentation.

The system developed in this project applied these principles through a web-based frontend with distinct *Training* and *Playing* modes. In *Training* mode, users could record and label gestures with minimal input. In *Playing* mode, motion data was continuously monitored, segmented automatically, and classified based on the trained templates. Communication with the backend and gesture recognition processing was handled transparently, requiring no user configuration. These strategies supported accessibility and usability, particularly for non-technical users. The field study with participants from Somos Nós showed that, with assistance, users were able to define gesture-sound mappings and engage meaningfully with the system, validating the effectiveness of these design choices.

6.1.3 How can the effectiveness of a gesture learning–based system be evaluated in terms of its impact on expressiveness, user satisfaction, and inclusivity?

The effectiveness of such systems can be evaluated through a combination of user observation, performance metrics, and inclusive field testing. Expressiveness is demonstrated when users can shape sound in ways that reflect their intent. User satisfaction can be assessed through levels of engagement and reaction to system feedback. Inclusivity is demonstrated by the system's ability to accommodate users with diverse physical or cognitive abilities.

In this project, expressiveness was observed through the variety of gestures users created and the control they had over the resulting sound changes. User satisfaction was evaluated through observation during the field study: participants were generally engaged and responsive, especially when gestures were recognized correctly. The field testing also highlighted areas where recognition failures impacted experience, underlining the importance of feedback consistency. Inclusivity was assessed by involving users from Somos Nós, who were able to participate in the training and use of the system despite varying motor and cognitive profiles. Their participation demonstrated that gesture-learning systems can function effectively in inclusive, community-based musical environments.

Building on these findings, it is equally important to acknowledge the system's limitations and challenges, which emerged during both field and lab-based evaluations. These factors provide critical context for interpreting the results and guiding future developments.

6.2 Limitations and Challenges

One of the primary challenges identified relates to the high variability of gestures performed by users, which often differ significantly from the simple geometric shapes originally envisioned (e.g., circles, lines, triangles). As discussed in the field study, this increased variability hampers the system's ability to generalize effectively, aligning with prior concerns regarding non-normative gesture input. This sensitivity exposes the need for improved gesture onset detection algorithms and more robust classification approaches that can better handle the unpredictability of user inputs, especially those with motor impairments.

Additionally, current recognition systems could benefit from enhanced feedback mechanisms during training, such as real-time confidence indicators and intuitive error correction tools, as highlighted in the evaluation of system limitations. These enhancements could improve user satisfaction and support more personalized, expressive interactions.

6.3 Recommendations for Future Work

Given these limitations, several avenues can be explored to improve the robustness and inclusiveness of gesture recognition systems in musical interfaces:

- **Enhanced Segmentation and Onset Detection:** Implement more sophisticated algorithms for detecting the start of gestures to reduce false recognitions and distinguish intentional gestures from unintentional movements [19].

- **Hybrid and Probabilistic Classification Models:** Combining different recognition techniques or employing probabilistic models can improve resilience to intra-user variability and environmental noise, leading to higher accuracy in real-world scenarios [22].

- **Few-Shot Learning and Continuous Adaptation:** Incorporating learning methods that require minimal training data and support ongoing adaptation by users can promote greater personalization and autonomy in gesture-based interaction systems [24].

- **Expanded User Testing and Subjective Evaluation:** Future studies should include larger and more diverse participant groups, assessing not only objective performance metrics but also users' perceived expressivity, satisfaction, and social inclusion aspects.

- **Integration into Broader Musical Ecosystems:** Linking the gesture recognition layer with other universal instruments and digital environments could expand expressive possibilities and foster more inclusive, participatory musical experiences [26].

6.4 Final Remarks

Although the current system does not fully eliminate the challenges associated with gesture variability or technical limitations, it signifies a meaningful step toward more inclusive and expressive musical interfaces. The findings reinforce that the success of accessible musical technology depends not only on technical optimization but also on understanding users' needs and contexts, centered on their agency and creativity.

This research underscores that gesture learning systems can be effectively integrated into community-based, participatory music tools that prioritize user empowerment and social inclusion. Moving forward, continuing to refine the system's robustness and user-centered design will be crucial to ensuring these tools can truly democratize musical expression across diverse populations.

References

[1] Music: A universal language. https://www.unesco.org/en/days/music. Accessed: 2025-05-18.

[2] Peter Beresford. Creating participatory public services: Participation, power and social learning. *Social Policy and Society*, 12(3):379–389, 2013.

[3] Peter Beresford. *All Our Welfare: Towards Participatory Social Policy*. Policy Press, 2016.

[4] Tony Bergstrom and Karrie Karahalios. Shaping expressive input for interactive music performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 657–666. ACM, 2007.

[5] John Blacking. *How Musical is Man?* University of Washington Press, 1973.

[6] Oliver Bown, Ross Bell, and Adam Parkinson. Accessible music interfaces: Design and evaluation. *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2017.

[7] Sam Bromley, Paul Vickers, and Antonella De Angeli. Playable environments for inclusive play: Evaluating an accessible digital musical instrument. *Journal of New Music Research*, 47(3):222–235, 2018.

[8] Rebecca Fiebrink, Dan Trueman, and Perry R. Cook. Real-time interactive supervised learning for music performance and composition. *Computer Music Journal*, 34(4):8–27, 2011.

[9] Emma Frid. Accessible digital musical instruments: A review of musical interfaces in inclusive music practice. *Multimodal Technologies and Interaction*, 3(2):57, 2019.

[10] Emma Frid and Roberto Bresin. The role of embodiment in the design of inclusive interfaces for music performance. In *Proceedings of the Sound and Music Computing Conference (SMC)*, 2018.

[11] William Gaver. What should we expect from research through design? *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 937–946, 2012.

[12] Niccolò Gillian and Joseph A. Paradiso. Adaptive and interactive machine learning techniques for musical gesture mapping. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2011.

[13] Mi.Mu Gloves. The mi.mu gloves for music interaction. https://mimugloves.com, 2018. Accessed: 2025-07-29.

[14] Andy Hunt and Marcelo M. Wanderley. Mapping performer parameters to synthesis engines. *Organised Sound*, 7(2):97–108, 2003.

[15] Alexander Refsum Jensenius and Michael J Lyons. A nime reader: Fifteen years of new interfaces for musical expression. In *Springer*, 2017.

[16] Sven Kratz and Michael Rohs. Combining acceleration and gyroscope data for motion gesture recognition using classifiers with dimensionality constraints. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, pages 173–178. ACM, 2013.

[17] Ronald L. Mace. The center for universal design: Universal design principles. https://projects.ncsu.edu/ncsu/design/cud/about_ud/udprinciples.htm, 1998. Accessed: 2025-05-18.

[18] Fabio Morreale and Perry R. Cook. Inclusive sound design: Designing for diverse abilities in music interaction. *Computer Music Journal*, 44(4):20–35, 2020.

[19] Martha C. Nussbaum. *Creating Capabilities: The Human Development Approach*. Harvard University Press, 2011.

[20] Graham Pullin. *Design Meets Disability*. MIT Press, Cambridge, MA, 2009.

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

[22] Atau Tanaka. Sensor-based musical instruments and interactive music. *The Oxford Handbook of Computer Music*, 2009.

[23] Sherrie Tucker, Pauline Oliveros, Leaf Miller, et al. Aumi: Adaptive use musical instruments. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2019. Available online.

[24] Marcelo M Wanderley and Marc Battier. Gestural control of music. In *Trends in Gestural Control of Music*. IRCAM, 2000.

[25] Marcelo M. Wanderley and Marc Battier. Gesture and music. In *Trends in Gestural Control of Music*, pages 1–23. IRCAM—Centre Pompidou, 2000.

[26] Kalin Yankov and Per Ola Kristensson. Protractor3d: a closed-form solution to rotation-invariant 3d gesture recognition. In *Proceedings of the 19th international conference on Intelligent User Interfaces*, pages 217–226. ACM, 2014.

[27] Norbert Zamborlin. Eyeharp: A gaze-controlled digital musical instrument. *Journal of New Music Research*, 46(2):171–181, 2017.

[28] John Zimmerman, Jodi Forlizzi, and Shelley Evenson. Research through design as a method for interaction design research in hci. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 493–502. ACM, 2007.