



Human-centric algebraic machine learning

Intelligent Tools for supporting creative professionals - Final

Deliverable D7.4

WP7 - Use cases



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952091.

Consortium



ALGEBRAIC AI



Chantalimaud
Foundation



EPROSIMA
The Middleware Experts



German
Research Center
for Artificial
Intelligence



TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

uc3m



Project

Title: ALMA: Human Centric Algebraic Machine Learning
 Acronym: ALMA
 Coordinator: Proyectos y Sistemas de Mantenimiento SL (eProsima)
 Grant agreement: 952091
 Call: H2020-EIC-FETPROACT-2019
 Program: Horizon 2020
 Start: 01 September 2020
 Duration: 54 months
 Website: <https://alma-ai.eu>
 E-mail: alma@eprosima.com
 Consortium: **Proyectos y Sistemas de Mantenimiento SL (eProsima) (EPROS)**, Spain
Champalimaud Foundation (CF), Portugal
ALGEBRAIC AI (ALG), Spain
German Research Center for Artificial Intelligence (DFKI), Germany
Inria (Inria), France
TU Kaiserslautern (TUK), Germany
FIWARE Foundation e.V. (FF), Germany
Technical Research Centre of Finland (VTT), Finland
Carlos III University (UC3M), Spain

Deliverable

Number: **D7.4**
 Title: **Intelligent Tools for supporting creative professionals - Final**
 Month: 54
 Work Package: WP7 - Use cases
 Work Package leader: Inria
 Deliverable leader: National Institute for Research in Digital Science and Technology (Inria)
 Deliverable type: R: Document, report
 Dissemination level: Public (PU)
 Date of submission: 2025-02-27
 Version: v2.0
 Status: Finished

Version history

Version	Date	Responsible	Author/Reviewer	Comments
v1.0	2024-09-28	Inria	Inria	draft
v2.0	2025-02-27	Inria	Inria, DFKI	final

Executive summary

AML Algebraic Machine Learning (AML) is a novel machine learning technique based on algebraic representations of data. Unlike statistical learning, AML algorithms are robust regarding the statistical properties of the data and are parameter-free. AML systems can retain what they learn, including sociocultural ethical practices for human–artificial intelligence systems. The aim of the EU-funded ALMA project is to study the mathematical properties of AML and to leverage them to develop a new generation of interactive human-centric machine learning systems. These systems are intended to minimize bias in order to facilitate trust while better enabling distributed collaborative learning.

This report corresponds to *Deliverable D7.4 - Intelligent Tools for supporting creative professionals - Final* of the ALMA project. This deliverable describes the design of *Tune Crafter*, an interactive music exploration system with dual support for pre-defined and user-defined gestures, catering to both beginners and advanced users. Its real-time feedback and AML-driven customization enable musicians to train and refine the system to their unique needs. This builds on our previous deliverable, D7.3 - Intelligent Tools for supporting creative professionals (Initial), and enables more advanced gesture training with AML.

This deliverable begins with a preliminary study to determine the current needs and limitations of music production using both digital and analog tools. It then expands on these insights in design implications and investigates them using design concepts. We describe the design and implementation of the working version of *Tune Crafter*, and demonstrate how users can use it to interact with music production by training an AML system to recognize gesture commands. The deliverable concludes with a brief summary and discussion of potential future research directions.

Contents

Executive Summary	3
Contents	4
Acronyms	5
1 Introduction	6
2 Background	7
2.1 Music Creation	7
2.2 Interactive Interfaces and Interfaces for Musicians	7
2.3 AI in Music	8
3 Preliminary Study and Design Implications	8
3.1 Methodology and Data Collection	8
3.2 Data Collection and Analysis	9
3.3 Results	10
3.3.1 Exploring rhythms and loops using existing sounds and musical objects:	10
3.3.2 Constructing music: channel by channel:	10
3.3.3 Exploring effects individually:	11
3.3.4 Visualizing effects helps to understand their impact on the tracks:	11
3.3.5 Providing feedback to enable trial-and-error approaches for refining sounds:	11
3.4 Design Implications	11
4 Prototype Design	12
5 Implementation	14
5.1 Interface Implementation for Fixed Gestures	15
5.1.1 Web Interface	15
5.1.2 System Architecture	16
5.2 AML for User Defined Gestures	19
5.2.1 Data Collection	20
5.2.2 Training data for AML	20
5.2.3 Discretize data	20
5.2.4 AML Models	20
5.2.5 Real time inference	21
6 Walk through	21
6.1 User scenario for Self Created Gestures	24
7 Discussion	26
7.1 Limitations	27
7.2 Future Work	27
8 Summary	27
References	29

Acronyms

AML Algebraic Machine Learning.

1 Introduction

We are particularly interested in creative application areas that aim to help both creative professionals and the general public engage in more creative expression through technology. In this use case for supporting creative professionals, we look into the possibility of using AML for music production.

Music production has undergone significant transformation with the rise of digital tools, which provide musicians with enhanced control over their compositions. At its core, music composition relies on the intuitive manipulation of sounds, rhythms, and effects. This research investigates how digital tools can enhance the creative process in music production, focusing on the interaction between musicians and these tools to shape soundscapes. The primary motivation is to understand digital music production and assist designers in bridging the gap between existing tools and user needs, while also exploring the explainability of AI tools to improve user experience.

This deliverable D7.4 - Intelligent Tools for supporting creative professionals (Final) begins with the question: How can musicians utilize digital tools to create rhythms? This involves thematic analysis and interviews with experienced musicians in digital music production. A music producer is defined as someone who creates music using digital tools, not just for manipulation but also for creation. Insights from these tools will inform design implications aimed at guiding the development of better tools, emphasizing the importance of rhythm manipulation, channel control, and iteration in composition. Music production is portrayed as a dynamic interaction between the composer and the tool, highlighting the significance of physical interaction, including gestures and hand movements, in shaping music. Such physical movements help musicians connect with their work, making the digital process more tangible and intuitive.

Additionally, the research underscores the importance of visual and auditory feedback in the music creation process. Musicians must use visual cues to make decisions during mixing or experimental phases, allowing them to discern which elements work well together and which require further development. The second research question addresses how to design a system for creating rhythms digitally, drawing insights from interviews with music production experts to inform design implications that enhance contemporary musicians' ability to compose innovative works.

The third research question focuses on the system's construction. After conceptualizing the design, the implications are implemented in the prototype system called *Tune Crafter*, which enables users to manipulate music through gestures. This system facilitates intuitive and expressive interactions in music production, allowing creative professionals to explore music combinations through gestures, a growing trend in human-computer interaction (HCI). Utilizing activity recognition techniques, including the MediaPipe Hand Pose detection model via TensorFlow, the system recognizes gestures through a three-dimensional hand model [1].

Tune Crafter operates in two scenarios: a fixed gesture system using a pre-existing library of MediaPipe's gesture dataset and a user-defined gesture system. The fixed system provides comprehensive guidance for various operations, while the user-defined system allows personalization by enabling users to train the algorithm with their gestures. Gesture classification is managed by Algebraic Machine Learning (AML) and other models, enhancing interactivity and accelerating user interaction cycles with the AML model. This framework aims to foster general ML literacy, enabling novices to train and explore Machine Learning approaches like AML, while also providing ML experts with a tool to evaluate creative possibilities in music production.

The following deliverable introduces existing work on the use case of supporting music production through interactive interfaces and machine learning models (Section 2). A preliminary study with 7 professional musicians helped identify the current needs of creative professionals using digital tools (Section 3), which is followed by our rationale for designing (Section 4) and implementing *Tune Crafter* (Section 5). We conclude with a brief discussion and summary of our work.

2 Background

2.1 Music Creation

Digital music production involves the use of digital audio workstations (DAWs) to record, edit, mix, and master audio in one place. Popular DAWs like FL Studio[2], Ableton Live[3], and Logic Pro X offer features for arranging musical elements, applying effects, and automating processes. These platforms support both MIDI and audio recording. Audio processing tools like FabFilter Pro Q3[4] and Valhalla Vintage Verb[5] are used to shape the sound within a composition. FabFilter Pro Q3 is an equaliser that adjusts the frequency balance of tracks, while Valhalla Vintage Verb provides reverb effects to add depth and space to the sound.

Loops and sampling are essential techniques in modern music production, especially in electronic and hip-hop genres[6]. A loop is a repeated sound section that can create rhythm or melody, and producers often manipulate these loops through pitch changes, time-stretching, or effects for variation. Sampling involves reusing portions of sound recordings in new tracks, allowing for the integration of diverse sounds and influences from various musical traditions.

Additionally, MIDI controllers and virtual instruments play a vital role in digital music production[7]. MIDI controllers enable real-time control of virtual instruments and software synthesizers within a DAW, enhancing expressiveness, while virtual instruments offer a broad array of sounds and textures for music creation[8].

Further, mixing and mastering are essential stages in music production, with mixing focusing on balancing individual tracks to create a polished sound, while mastering optimizes the overall sound for various playback systems[9]. Equalization and filtering allow producers to further control a track's frequency content, shaping its tonal balance and ensuring clarity in the mix, see [10]. Advanced effects, such as slicer, looper, and reverb, also play an increasingly crucial role in sound design. For DJs for example, seamless track mixing is vital, with features like sync, hot cues, and beat navigation enhancing performance capabilities[11].

2.2 Interactive Interfaces and Interfaces for Musicians

The evolution of human-computer interaction (HCI) has seen a significant shift towards immersive and intuitive interfaces, as highlighted by Weiser's vision of technology that seamlessly integrates into daily life [12]. Apple's Vision Pro exemplifies this concept of Spatial Computing, where users engage with applications and data in their environment through natural gestures, making the technology less intrusive and more focused on the task at hand. This transition from traditional monitors to augmented reality (AR) interfaces has been propelled by advancements in HCI and design, particularly through the development of Tangible User Interfaces (TUIs). Ishii and Ullmer's foundational work on Tangible Bits emphasizes the integration of digital and physical realms, critiquing the limitations of graphical user interfaces (GUIs) that fail to utilize the richness of human sensory experiences [13].

Further research by Nancel et al. explores gesture interaction techniques, revealing that mid-air gestures can be less efficient than device-based inputs but can be optimized through design factors such as guidance and movement types [14]. This work aims to enhance user experiences in digital music environments by applying these findings to gesture-based music production systems. Tsandilas et al. introduce "Knotty Gestures," which allow users to interact subtly with their writing, creating opportunities for immediate and future interactions [15]. This reflects a broader trend in digital music where musicians seek intimate control over their instruments, as discussed by Wessel and Wright, who emphasize the importance of low latency and expressive control in musical gestures [16].

Mackay's exploration of customizable software presents a co-adaptive relationship between users and technology, where both influence each other over time [17]. This dynamic is crucial for understanding

user behavior and software development, especially in the context of musical systems. Recent work by Peng et al. on DesignPrompt illustrates how multimodal interaction can enhance design exploration and self-expression, aligning with the principles of intuitive and flexible creative expression [18]. Finally, Blaine and Fels highlight the significance of collaborative musical experiences, advocating for systems that support group interaction and real-time feedback, thereby fostering creativity in musical performance [19]. Together, these studies underscore the ongoing evolution of interaction paradigms that prioritize user engagement and collaborative creativity in digital environments.

2.3 AI in Music

The MediaPipe Hand Pose detection model, accessible via TensorFlow [20], is utilized in this work to recognize hand gestures through a 3D hand model [1]. This model employs a straightforward algorithm to calculate gestures based on three distinct datasets: an in-the-wild dataset, an in-house collected gesture dataset, and a synthetic dataset. It produces three outputs: 21 hand landmarks with x, y, and relative depth coordinates, a hand flag indicating the probability of hand presence, and a binary classification of handedness (left or right hand). The algorithm begins by determining the state of each finger (bent or straight) through accumulated joint angles, which are then mapped to predefined gestures, allowing for the estimation of basic static gestures with reasonable quality.

In the realm of Interactive Machine Learning (IML), Caramiaux (2023) [21] highlights its significance in enabling rapid and incremental model updates in response to user input, fostering a continuous dialogue between users and machine learning systems. This interaction is particularly vital in creative practices, where subjective assessments by artists and designers are essential. Baptiste's doctoral research delves into gestural interfaces and musical expression, exploring how 3D hand movements can intuitively control musical parameters through variations in gesture velocity and trajectory. The concept of machine learning as a design material emphasizes exposing the internal workings of these systems, allowing users to manipulate and modify them for creative exploration. This inductive approach to programming by demonstration has made machine learning more accessible to non-experts in the creative industry, enabling innovative motion-sound interactions in artistic performances.

Further, Sun et al. (2022) [22] conducted workshops with software engineers to identify explainability needs in generative AI for code through a scenario-based design approach. Their findings, which categorized 11 explainability needs, inform the thesis aimed at enhancing human-AI collaboration. Understanding user engagement with these systems and the feedback received is crucial for improving overall experiences. Fiebrink et al. (2016) [23] also emphasize the importance of explainability in music recommender systems, suggesting that transparent algorithms can enhance user trust and satisfaction by not only providing accurate song suggestions but also explaining the rationale behind them. Lastly, Bryan-Kinns et al. (2023) [24] focus on familiar interface elements for musicians, emphasizing real-time feedback between musicians and instruments, with algebraic machine learning serving as the xAI component to enhance user understanding of system operations.

3 Preliminary Study and Design Implications

3.1 Methodology and Data Collection

To explore the experiences and workflows of end-users that are involved in music production, we conducted a number of semi-structured interviews. A total of seven participants, all with backgrounds in music production, mixing instruments, and live performance, were recruited through electronic means, specifically social media platforms such as Instagram and Facebook, as well as through word of mouth. To qualify for participation, individuals were required to have a background in playing music and knowledge of mixing or producing music using either tactile turntables or digital mediums. Demographic data,

including age, gender, choice of musical instruments, and years of experience in music, were collected from each participant.

The interviews were conducted individually, with each session lasting between 45 minutes to 1 hour and 15 minutes. The interview process began with an open-ended question regarding the participants' workflows for creating rhythms. Participants were then prompted to elaborate on their methods for discovering new sounds and rhythmic patterns, including their criteria for selecting sounds for their projects. Further inquiries focused on the tools and procedures used to modify existing rhythms.

Participants were also asked about their experiences with digital tools, such as drum pads in virtual environments or touch-sensitive surfaces, which aim to replicate physical interactions. They were encouraged to compare the ease and difficulty of using these digital tools versus traditional physical instruments. The discussion included the types of feedback—visual, auditory, and haptic—that participants found most beneficial in their creative processes, as well as the impact of rhythm visualizations, such as waveform displays or beat grids, on their workflows.

Additionally, the interviews explored how physical actions and experiences influenced the participants' creativity and music expression, including the naturalness of different movements and any challenges they faced. Participants were asked about tasks they routinely performed when generating rhythms and those they engaged in less frequently, along with the reasons for such variations. Finally, the study investigated how participants modified their workflows and tools in music creation, focusing on the aspects they frequently changed and the motivations behind these customizations.

ID	Age	Gender	Choice of Musical Instruments	Choice of Digital Tools	Years of Experience
P1	23	Male	Piano	FL Studio	3
P2	24	Male	Guitar	Rekordbox & Pioneer DDJ 1000	5
P3	29	Male	Guitar	FL Studio	6
P4	29	Male	Guitar	FL Studio &	10
P5	24	Male	Guitar	FL Studio & Rekordbox	5
P6	22	Male	Drums	Abelton & FL Studio & Rekordbox	7
P7	25	Male	Guitar	FL Studio & Audacity	10

Table 2: Demographic and Tool Usage Data of Participants

3.2 Data Collection and Analysis

The data collected from these interviews were analyzed using thematic analysis, which allowed for the identification of key themes and patterns in the participants' responses. The findings were subsequently reflected upon to provide recommendations for the design of music technology that could enhance the structuring of music production and playback processes.

In our study, informed consent was obtained from all participants through a consent form, which included permission for audio recording during the interviews. Before conducting the interviews, participants were fully briefed on the purpose of the study and their rights, ensuring ethical compliance. All audio recordings were subsequently transcribed using a GDPR-compliant whisper tool to safeguard participant privacy.

Our data analysis used a combination of bottom-up thematic analysis approach, where each statement

was coded and then clustered based on themes in the data. We grouped similar codes to form high-level themes.

3.3 Results

The following section presents our findings from the interview study. The findings are split into five themes: (1) exploring rhythms and creating music using existing sounds and musical objects; (2) constructing music: channel by channel; (3) exploring effects for each channel individually before merging them; (4) Visualizing effects helps to understand their impact on the tracks; (5) Providing feedback to enable trial-and-error approaches for refining sounds.

3.3.1 Exploring rhythms and loops using existing sounds and musical objects:

A common quality shared by all participants (7/7), is their natural exploration of existing sounds and rhythms to create new musical pieces. The majority of participants underlined the importance of employing pre-existing loops, samples, and rhythms as the composition's backbone. For example, P5 proposed starting with an existing sample or loop and then building around it to create something new. Similarly, P1 and P3 stated that they utilize their sound collections to locate sounds and create loops as a starting point for new tunes. This suggests that musicians frequently go to what is known in order to expand on it in a unique way. P1 discussed layering songs, starting with a basis and adding different instruments. P2 built on the ability to record numerous loops from various sources and toggle them on and off as required. P3 emphasized the necessity of using loops to organize songs and change them live, resulting in a unified musical structure inside the interface. P4 discussed adding numerous materials to a basic structure, such as orchestral pieces or vocal tracks, and then refining them with tools like SoundCloud and audio equalizers. P5 stressed the traditional methodology of utilizing loops to create transitions, while also experimenting with other ways such as scratch transitions. P6 mentioned employing loops to generate specific effects like hi-hats, and P7 emphasized the need of adding tiny touches, such as orchestral embellishments, to influence the overall composition.

3.3.2 Constructing music: channel by channel:

The majority of participants (6/7) stated that music production is primarily a layered technique, in which songs are built progressively by combining various instrumental and auditory parts to make a finished work. This method was broken down into two sub-themes: (1) track layering and (2) song structure.

Track Layering Process: Producers often begin songs with a fundamental layer (e.g., rhythm or melody) and gradually add instruments or features to increase complexity. Participants experimented with recycling material from previous pieces, testing the compatibility of pre-existing loops or themes with new works. For example, P1 discussed starting with a "base" layer and progressively adding instruments, but P2 emphasized the flexibility of recording several loops from different sources and dynamically switching their activation. P6 and P7 go on to discuss more complex layering approaches, such as employing loops to generate particular effects (e.g., hi-hat patterns) or stacking synthesizers to create orchestral sounds that add depth to the song.

Structuring the Song: Participants stressed the need of using digital interfaces to organize layered materials into cohesive arrangements. This includes separating sections (e.g., intro, verse, chorus) to maintain structural clarity. P3 highlighted real-time loop tweaks to improve transitions and preserve flow, whereas P5 explored using loops for transitions while also experimenting with alternatives such as scratch approaches. P4 described how they enhanced a core framework with new parts (e.g., orchestral layers, voice recordings), using technologies like SoundCloud, instrument libraries, and audio equalizers to polish

Overall, participants frequently emphasized the need of iterative refining in their approach. Techniques spanned from P7's carefully synthesizer layering to P4's integration of external audio tools, displaying both shared methodology (e.g., loop-based structure) and unique approaches (e.g., scratch transitions).

3.3.3 Exploring effects individually:

All participants (7/7) indicated that producers use numerous effects to shape and improve the sound of particular channels or instruments, which can significantly change the overall vibe of the track. P1 talked about experimenting with effects to see how they modify the sound, saying that the appropriate effect may greatly alter a track's feel. P2 discussed the creative use of effects such as reverb to accentuate particular areas of a song. P3 did not explicitly discuss effects, but did imply their significance in the larger context of music production. P4 stressed that effects should be tailored to user preferences in order to get the desired sound. P5 emphasized that effects are critical for establishing the correct mood in a track, particularly when employing reverb to widen voice sounds. P6 detailed using digital interfaces to apply effects to various instruments and channels, while P7 discussed experimenting with different effects on symphonic works to perfect the sound. Types of Effects: Common effects include reverb, compression, and equalization. These may be used to enhance sounds, change frequency ranges, and clean up vocals.

3.3.4 Visualizing effects helps to understand their impact on the tracks:

The majority of participants (5/7) emphasized the importance of visualization in applying and changing sound effects during production, stating that visualizing effects helps them understand their impact on the track, as well as leading to more informed adjustments. P2 emphasized the significance of visual indications in organizing live performances, particularly in defining frequency ranges. P4 stressed the importance of viewing frequency charts in establishing an instrument's most prominent zones, which makes it easier to balance the mix. P5 explained how visual input on a mixing table, such as BPM displays, improves the accuracy of changes. at the same time, P6 stated that both visual and tactile input are essential for creating appropriate sound modifications, whereas P7 emphasized the importance of visual feedback in receiving and integrating external feedback throughout production.

3.3.5 Providing feedback to enable trial-and-error approaches for refining sounds:

Most participants (6/7) highlighted the value of feedback in the iterative process of music development. P1 emphasized the importance of system feedback in modifying controls, adopting a trial-and-error approach to fine-tune the final sound. P4 highlighted on how visual and audible feedback are necessary for making exact changes, especially when working with virtual instruments and effects. P5 and P6 both emphasized the significance of tactile feedback when playing instruments, implying that this feedback loop is critical for keeping control over the production processes. P7 also emphasized the advantages of digital interfaces, which allow a high degree of control and are vital for experimenting with effects in real time and modifying recordings based on input from others.

Three of the participants (3/7) further discussed the learning curve and customisation of their systems. Several users for example color-coded their channels and grouped them into groups to play music. Some participants suggested that gestures could be used instead of clicks for numerous activities since they are more intuitive, but not have encountered systems that would allow it.

3.4 Design Implications

These insights resulted in the following four design implications for designing music creation systems:

D1: Adaptive and Customizable Interfaces for Intuitive Music Creation Participants highlighted significant challenges with current tangible and haptic music creation systems, specifically point-

ing out issues like a lack of adaptability, high complexity, and steep learning curves. Many noted that physical controls, such as knobs, are often not intuitive and can be difficult to adjust, complicating the adaptation to new or changing use cases. To address these constraints, participants emphasized the importance of offering easily definable and customizable input methods that give users full control. They expressed a strong need for customization and the ability to learn new systems, underscoring the necessity for adaptive interfaces. An adaptive system would learn and adjust to users' behaviors and preferences over time, leading to a more personalized and efficient interaction experience. This would allow for simpler and more effortless sound, effect and track manipulation depending on the users need while minimizing dependence on physical controllers and existing interfaces.

D2: Intuitive Layering and Dynamic Compositional Flow Participants emphasized the importance of systems providing features that facilitate the seamless addition, modification, and organization of layers and tracks. This capability is crucial for musicians as it allows them to explore a wider range of creative possibilities while maintaining coherence in their compositions. They noted that a typical composition is structured into an introduction, body, and conclusion (outro), and highlighted the technique of gradual buildup. This technique involves incrementally layering small sound elements throughout the different sections of a composition, enabling musicians to effectively construct and develop their work. Systems that support music creation should hence facilitate intuitive layering and structuring of sound elements to support gradual build of such compositions.

D3: Simplifying Cross-Channel Customization to Reduce Learning Resistance Participants expressed concerns regarding the learning curve and customization of their systems, noting that while current systems facilitate cross-channel work, the interactions can be complex and difficult to master. They highlighted the challenges posed by varying effects and characteristics across different channels, such as musical instruments, which necessitate different levels of adjustments. To address these issues, there is a significant opportunity to develop simpler, more user-friendly interactions that enable seamless operations across channels. By enhancing customization options and streamlining the learning process, the system could reduce resistance to learning and foster greater adoption among users.

D4: Integrating Visual-Tactile Feedback for Real-Time Sound Control Participants emphasized the importance of visualization in sound production, highlighting the need for visual feedback to accurately understand and adjust sound effects. They suggested incorporating features such as BPM (Beats Per Minute) indicators, waveform displays, and the separation of audio elements like vocals, instruments, and bass, along with frequency charts. These visual tools would provide clear, real-time feedback, empowering users to make informed decisions during production. Additionally, participants noted the value of physical tools like knobs, sliders, and touchscreens, which offer tactile feedback for precise adjustments through direct interaction for some of these adjustments. Therefore, it is crucial for systems to enhance their visualization capabilities to support real-time adjustments.

4 Prototype Design

A brainstorming session led to the development of a design concept for an interactive music production system, focusing on addressing design implications D1 and D2. This concept features two primary approaches:

1. **Fixed Gestures:** Users can utilize predefined gestures that are mapped to specific functions within the system.
2. **User-Created Gestures:** Users are empowered to design their own gestures and map them to the system, enhancing personalization.

To explore gesture interaction for creative professionals, various AI models can be employed, and our work aims to evaluate differences in user experience and system performance. The emphasis is on specific interaction use cases in music production that could significantly enhance digital music creation.

Several potential solutions emerged from the brainstorming session, which were relevant for our future use cases: 1. **Gesture-Based Interactions:** Simple and straightforward motions for numerous music production components including channels, loops, and effects.

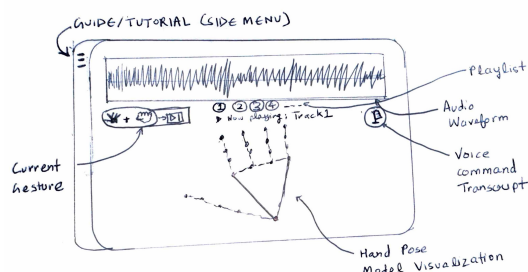
2. **Channel Management and Layering:** Methods for organizing and managing music layers and tracks to ensure a smooth composing process.

3. **AI Integration for Sound Manipulation:** Using AI to help with tasks such as gesture recognition, explainable AI, and creating new sound parts.

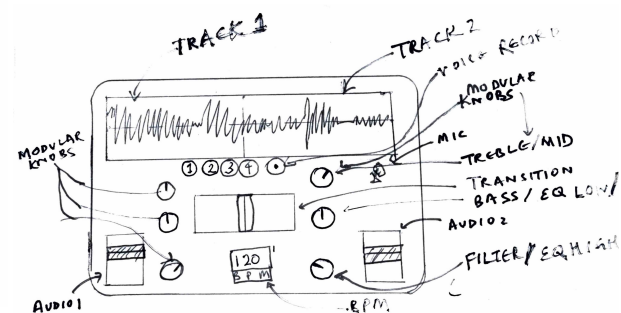
4. **Visualization and Feedback Mechanisms:** Improved visual and audio feedback to help users understand the consequences and modifications made to their compositions.

5. **Customization and Adaptability:** Ensure that the system enables customizable processes and responds to unique user preferences.

We found that certain capabilities were more important than others for a music adaptation system to work well on a variety of use cases and music styles. The system's main user interface is seen in Figure 1a. We also looked for inspiration for key characteristics by analyzing professional DJ equipment and digital audio workstations (DAWs), as well as their interaction approaches. Building on these findings, we investigated an alternative system's user interface, which was meant to promote simple and adaptable interactions known to many music producers (seen in Figure 1b). However, we identified a number of limitations due to the crowded interface space and fixed features represented by virtual buttons. As a result, we simplified the interface in order to fully exploit the capabilities of gesture exploration in musical composition, as suggested by our initial sketches.

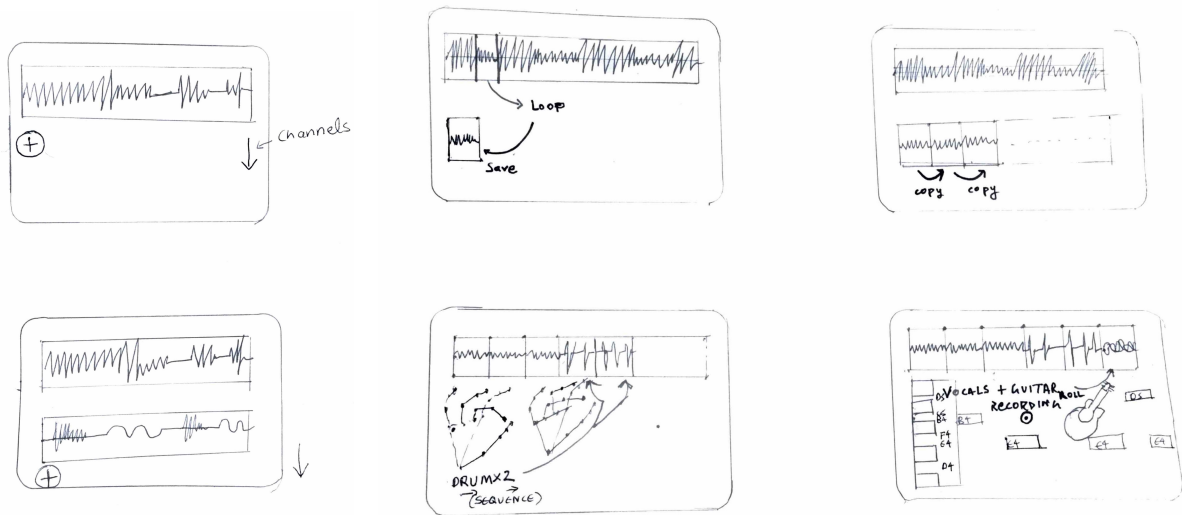


(a) Initial sketch of the main interface, highlighting potential visualization approaches for effects, gesture input and navigations



(b) Alternative sketch of the main interface inspired by DJ sound mixers

Users may add and modify numerous channels using the multi-channel interface, which is seen in Figure 2a. It draws on the design characteristics and implications that were explored in earlier brainstorming sessions. A storyboard drawing that illustrates how to compose, record, and work with live music components is shown in Figure 2b. The storyboard progresses from left to right, beginning with the first and then second row: First, the user uses motions to choose a loop, and then uses another gesture to copy it. After that, they use gestures to play percussion patterns, and to finish the song, they record voices using an external visual plugin for guitar bands. The hands-free interaction approach shown in the storyboard enables users to carry out all tasks using gestures.



(a) Sketch of potential multi-channel systems to compare, align and merge different channels

(b) Storyboard Sketch explaining how various interactions can help in Loop, Copy and other operations

5 Implementation

The design concept was implemented in *Tune Crafter* to demonstrate its practicality in an interactive scenario. A web application was created using modern UI toolkits and Google Media Pipe for creating tunes. The technical aspects, including frontend and backend development, will be described below.

5.1 Interface Implementation for Fixed Gestures

5.1.1 Web Interface

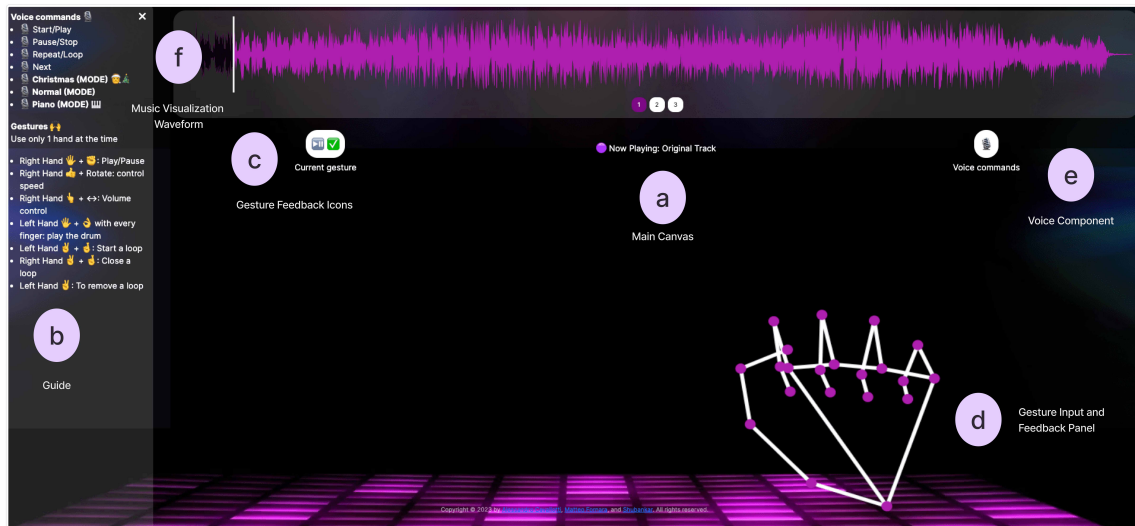


Figure 3: Main interface of *Tune Crafter*

The user interface shown in Figure 3 is designed to provide musicians with an intuitive and comprehensive workspace for music creation and manipulation. It consists of several key components, each tailored to enhance the user experience and streamline the creative process:

(a) **Main Canvas:** The Main Canvas is the central workspace where musicians can arrange, edit, and manipulate tracks, loops, and effects. This area serves as the primary hub for composing and organizing musical elements, offering a flexible and dynamic environment for creative exploration.

(b) **Guide:** Adjacent to the workspace is the Guide, a visual mapping of gestures to specific commands or music operations. This feature ensures that users can easily understand and apply gesture-based controls, reducing the learning curve and enabling seamless interaction with the system.

(c) **Feedback Icons:** To assist in precise sound manipulation, the interface includes icons that provide real-time visual feedback. These icons display critical information, such as waveform representations, helping users make informed adjustments to their compositions.

(d) **Gesture Input and Feedback Panel:** This panel allows users to create, edit, and apply gestures to various music elements. This panel also provides a live view of the user's hands and gestures, ensuring accurate and responsive control over the system's functions.

(e) **Voice Component:** This component enhances accessibility by transcribing live audio into commands and displaying them as text. This feature enables hands-free operation and supports users who prefer voice-based interactions, further expanding the system's versatility.

(f) **Music Visualization Waveform:** Finally, the visualization panel displays the waveform of music playback, offering a clear and detailed representation of the audio. This visualization aids in monitoring and fine-tuning the composition, ensuring that users can achieve the desired sound with precision and could be adapted to any user preferred visualization methods.

Together, these components create a cohesive and user-friendly interface that empowers musicians to experiment, create, and refine their work with ease. By integrating gesture-based controls, real-time feedback, and versatile input methods, the system bridges the gap between creativity and technology, providing a robust platform for modern music production. The available mapping of each gesture command to music operation is shown in Table 3.

Interaction Technique	Main Feature
Right Hand (open hand) + (fist)	Play/Pause
Right Hand (thumbs up) + Rotate	Control speed
Right Hand (pointing up) + (left-right arrow)	Volume control
Left Hand (open hand) + (OK sign) with every finger	Play the drum
Left Hand (peace sign) + (crossed fingers)	Start a loop
Right Hand (peace sign) + (crossed fingers)	Close a loop
Left Hand (peace sign)	To remove a loop

Table 3: Gestures and their corresponding main features.

5.1.2 System Architecture

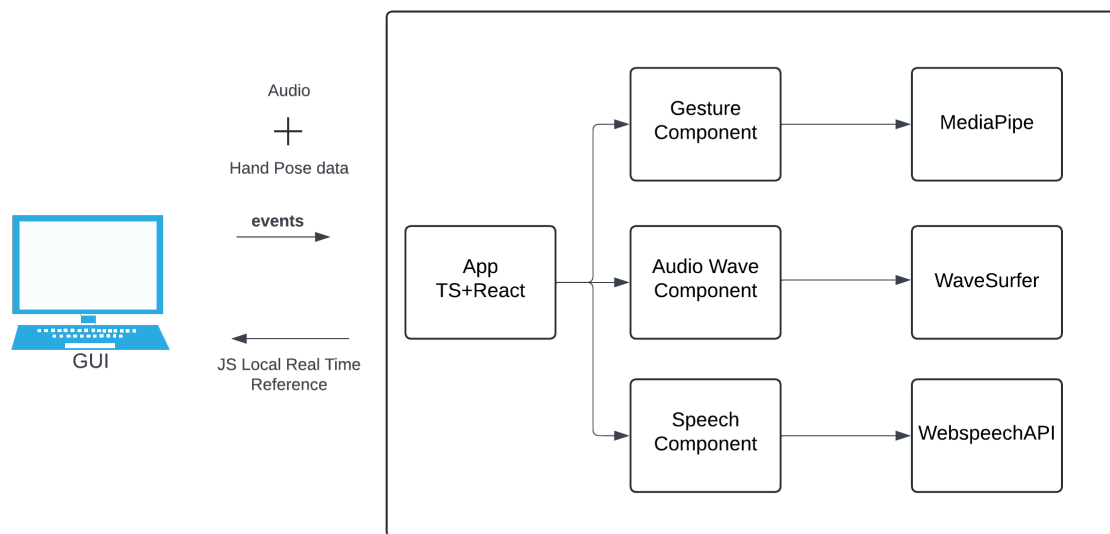


Figure 4: System Architecture Overview of the Fixed Gesture Implementation

Dependencies:

1. *React*: Utilize React for building user interfaces efficiently.
2. *WaveSurfer.js*: WaveSurfer.js is used for audio waveform visualization and playback control.
3. *MediaPipe*: MediaPipe is used for real-time hand gesture recognition.
4. *Bootstrap*: Bootstrap CSS is employed for styling and layout purposes.

The following description refers to the code available on Github at : <https://github.com/shubhankarx/Tune-Crafter>

I. MainComponent.tsx/ Application The Application serves as the core component of the system, responsible for rendering and coordinating all other elements to ensure seamless functionality. Built using React for its robust user interface capabilities and TensorFlow for advanced machine learning integration, this component forms the backbone of the system, enabling dynamic and responsive interactions. Key Interactions with other components are:

Gestures and Speech: The system leverages the GestureComponent and SpeechComponent to facilitate intuitive user interactions. These components work in tandem with the waveform, webcam stream, and microphone to enable gesture-based controls and voice commands. For instance, gestures captured via the webcam or voice inputs through the microphone are processed and translated into actionable commands, allowing users to manipulate audio elements effortlessly.

Audio: The AudioWaveComponent is responsible for rendering and interacting with the audio waveform. It provides real-time visual feedback, enabling users to monitor and adjust audio playback with precision. This component ensures that users can seamlessly edit and manipulate sound elements, enhancing the overall creative experience.

Webcam: The webcam is initialized automatically when the component mounts, provided the browser supports it. This feature enables gesture recognition and real-time visual feedback, ensuring that users can interact with the system naturally and effectively.

II. AudioWaveComponent.tsx The AudioWaveComponent is responsible for visualizing audio waveforms, offering users a clear and interactive representation of their audio files. Built using the **WaveSurfer** utility from the **wavesurfer-react** library, this component renders a detailed waveform visualization that can be controlled and interacted with from its parent component. This visualization allows users to monitor audio playback, identify specific sections, and make precise adjustments, making it an essential tool for audio manipulation and editing.

III. VolumeProgressBar.tsx The VolumeProgressBar component provides real-time visual feedback on audio volume or amplitude levels. It displays a progress bar where the width corresponds to the volume percentage, enabling users to quickly assess and adjust audio levels. This intuitive visual representation ensures that users can maintain optimal audio balance and make informed adjustments during playback or recording.

IV. GestureComponent.tsx The GestureComponent introduces a hands-free, gesture-based interaction model, allowing users to control audio and video playback dynamically. It integrates with the **GestureRecognizer** from the **@mediapipe/tasks-vision** library to detect and interpret hand gestures. These gestures are then mapped to specific functionalities, such as play/pause, volume adjustments, and the application of special effects. Additionally, the component supports voice commands, further enhancing its versatility and accessibility. By combining gesture and voice inputs, this component provides a seamless and immersive control experience, reducing reliance on traditional input methods and enabling more creative freedom. In Figure 5 we show examples on how the states and actions that the module can take are defined.

V. GestureModel.tsx The GestureModel component defines the data model for gestures and their associated labels. It serves as the foundation for the gesture recognition system, providing the structured data that the GestureComponent relies on to interpret and map user gestures to specific actions. By organizing gestures into a clear and consistent model, this component ensures accurate and reliable gesture recognition, enabling users to control audio and video playback dynamically through hand movements.

VI. SpeechComponent.tsx The SpeechComponent is responsible for recognizing voice commands and translating them into actionable tasks. Utilizing speech recognition technology using the Wenspeech API,

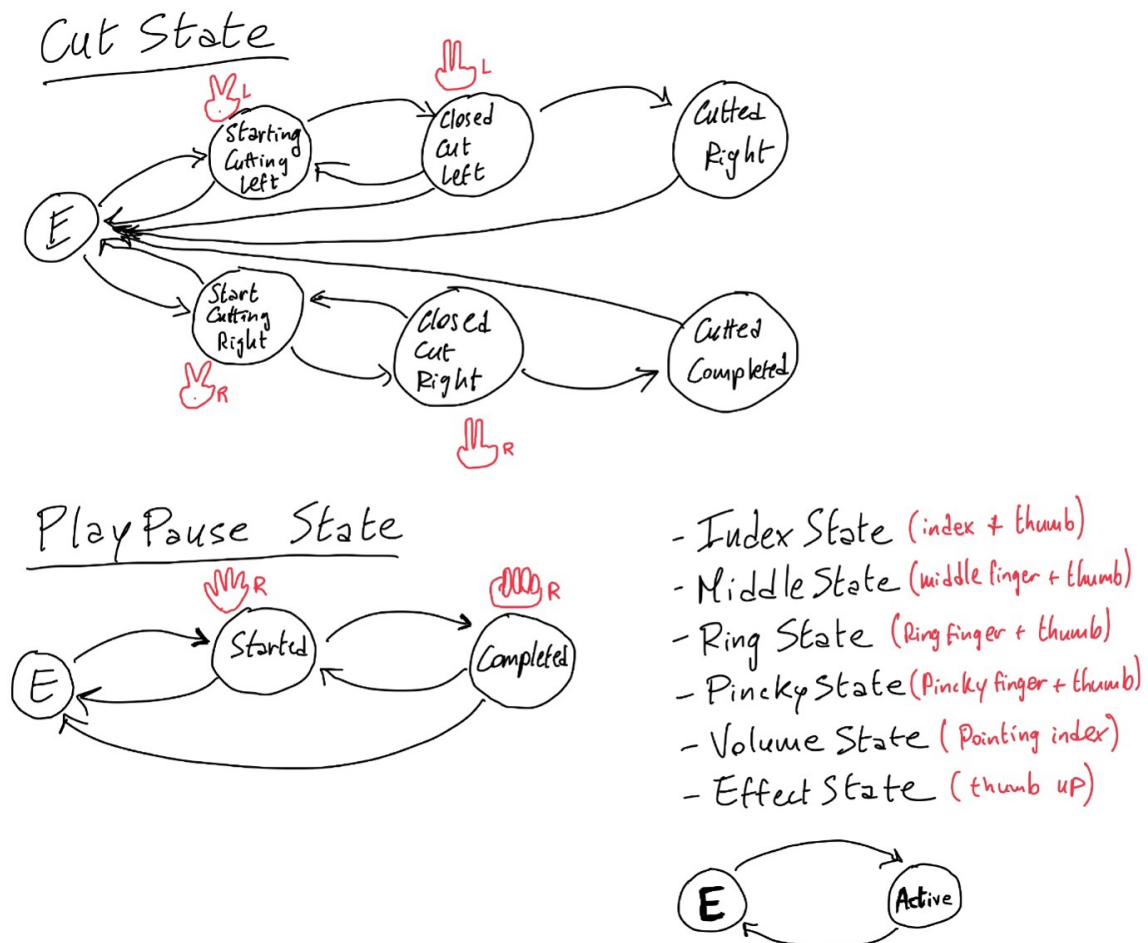


Figure 5: Finite state machines architecture in *Tune Crafter* to enable seamless transitions between various gesture states

this component detects spoken words and triggers corresponding actions, such as play/pause, volume adjustments, or the application of effects. By integrating voice control, the system enhances accessibility and provides users with an alternative, hands-free method of interaction, further streamlining the creative process.

VII. *AudioManager.tsx* The AudioManager component handles the loading and management of audio files, ensuring smooth playback during web code execution. It initializes the audio context for each sound and stores the audio data in a buffer for quick recall when needed. This component is essential for maintaining audio performance and responsiveness, allowing users to manipulate and play back audio files without interruptions. Its efficient management of audio resources ensures a seamless experience, even when working with multiple tracks or complex compositions.

5.2 AML for User Defined Gestures

Our implementation took place in three stages: The first is the user interface (see Figure 6) that handles data collection and processing developed using the same user-defined gesture approach system, as already described in D4.8 - Working Prototype and Designs (Revised). The second part is generating trained AML class models from the training examples provided by the user. Our AML server was implemented in Python using Flask, as the AML engine has Python bindings. The model structure is then serialized to JSON and sent to the web application as illustrated in Figure 7. For the comparison algorithms we used tensorflow-js, whose models can be easily integrated into our codebase, allowing the user to compare AML results with a large number of available models.

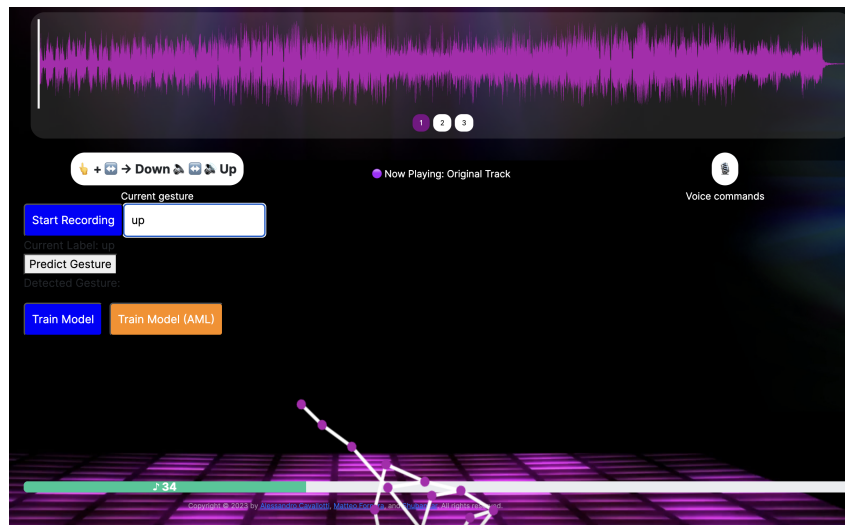


Figure 6: AML data collection in *Tune Crafter*

In the third part, the system maps personally defined gestures to sound patterns using real-time prediction based on the selected model.

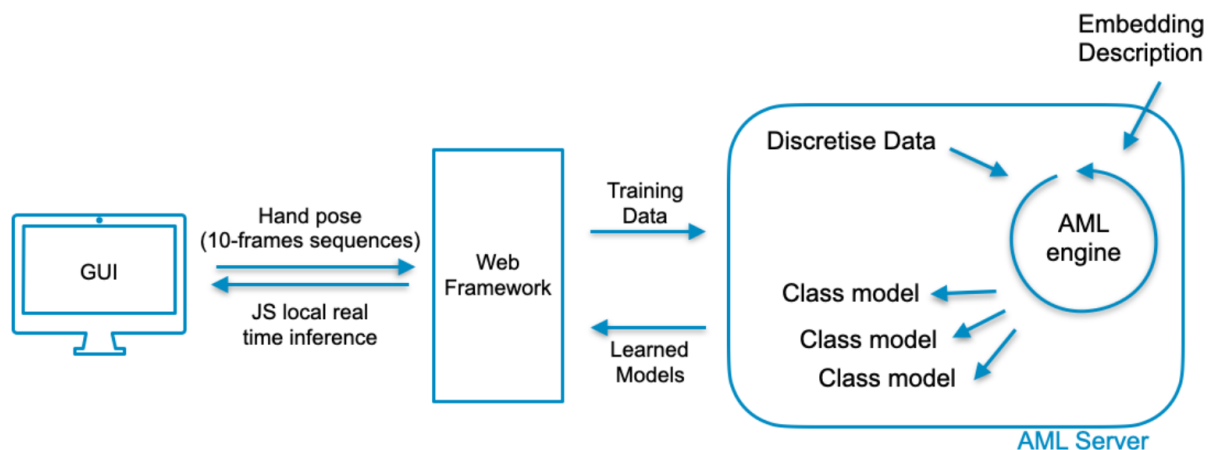


Figure 7: System Architecture Overview of the User-Defined Gesture Implementation

5.2.1 Data Collection

Data is collected via the user's camera. A real-time hand posture estimation model [25] processes images of the user's hand joints in the browser. We utilize the MediaPipe Hand Pose detection model from Tensorflow [20]. In D4.8 - Working Prototype and Designs (Revised), we employed one hand - one point features based on picture coordinates and index finger 3D position. This limited the range of gestures that users could create and apply. In *Tune Crafter*, we employ the 3D locations of all 5 finger tips, allowing users to construct complex gestures in mid-air.

We employ 10-frame hand posture coordinate sequences to produce gestures, which imply movement. After the user starts collecting motion data, each ten-frame sliding window produces one instance. Thus, while data collection is in progress, we build instances using a 10-by-1 frame. In our previous deliverable, we used the (x, y, z) location as a method to categorize the gesture over time. However, we observed that this limited gestures to certain regions, making it ambiguous for end users. Instead, we defined our fixed coordinates as differences between the position of the finger at the start of the recording $t_i = (x_i, y_i, z_i)$ and the following recorded frame $t_{i+1} = (x_{i+1}, y_{i+1}, z_{i+1})$. This allows us to become independent of the gesture's exact position, giving the end user additional flexibility of movement.

5.2.2 Training data for AML

Each step in our 10-frame window yields 15 values. For each of the user's five fingers, we gather the $(x_{diff}, y_{diff}, z_{diff})$ coordinates in the hand coordinate system as retrieved by [25]. This implies that each instance is a vector of size 1500, with the first 15 values for the hand position identified in the first frame of the sequence, the second 15 representing the values in the second frame, and so on. If the hand is not recognized at a certain frame in the sequence, the 15 associated values are set to zero.

5.2.3 Discretize data

Our example continues to use the engine version that can only handle discrete data, because the continuous version increases linearly with each new detected value. In the AML Server, each feature is discretized independently, using [26] to find the best number of bins from the training data, as done in the D4.8 - Working Prototype and Designs (Revised). The bin edges will be returned to the Web Framework later, along with the learnt class models, as they are required for real-time prediction in the browser.

5.2.4 AML Models

Similar to D4.8 - Working Prototype and Designs (Revised), the AML server returns a JSON representation of the atoms for each class model, along with a description of discretization bins. This allows us to convert any real value in real time to its bin and receive the algebraic constants associated with that bin. We can determine the number of atoms present for each class by adding all instance constants and then predicting the class based on the number of missing atoms. In the ideal instance, a positive sample of a class should have no missing atoms. In practice, we give classification outputs and measure confidence by calculating the total number of misses in the training set's negative instances for each class. In more formal terms, the score a test example x has for class k is

$$1.0 - P(\geq m_k \mid -)$$

where m_k is the observed number of misses for x based on the model, k is the positive class, and $P(\cdot \mid -)$ is the probability of misses computed solely over negative training samples of class k . The confidence values are then calculated by standardizing confidence ratings across classes.

5.2.5 Real time inference

The live prediction of trained hand gestures is based on the camera stream, which is sent to the hand detection pipeline and trained models. These models are loaded in the browser during the training step and run locally. In the case of AML, the data is discretized based on bins for each feature, obtained constants, and determined the number of atoms present for each class. This is implemented in the browser in Javascript using set operations. The confidence level is then mapped to the volume of a pre-determined audio track, played in loops to allow a smooth continuous stream of sound exploration. The system can be retrained for different instruments or gesture changes.

6 Walk through

The following scenario illustrates how Max, a DJ and musician would use *Tune Crafter* to play music.

Track Selection and Playback

Max begins by speaking the name of the track he wants to play. The voice detection system picks up the track. He then selects multiple tracks and eventually settles on a techno track. Max initiates playback by performing a closed fist action, which the system recognizes to start the track.

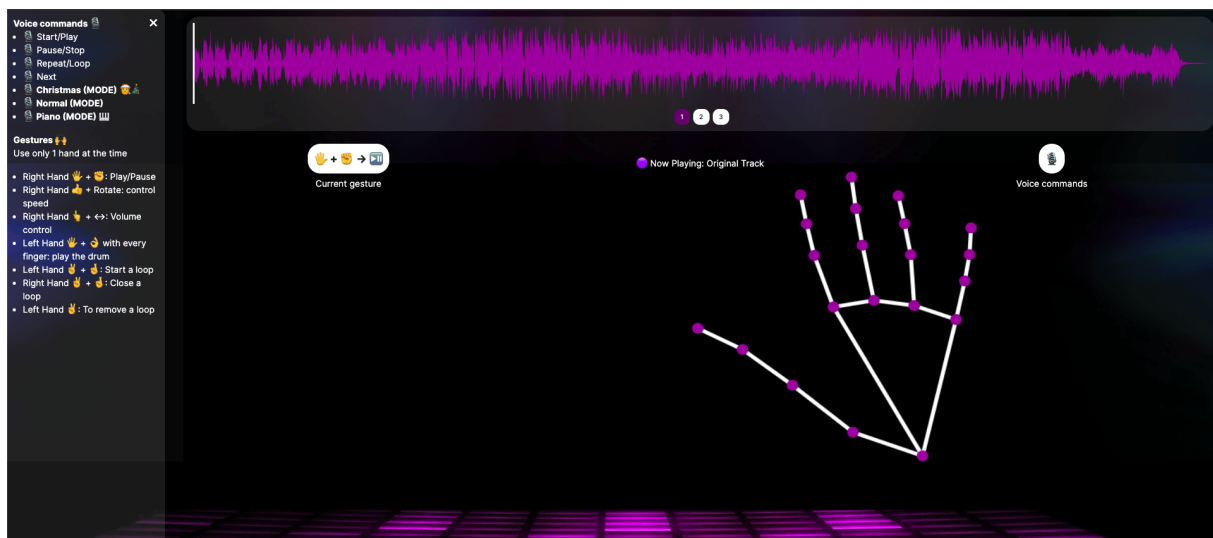


Figure 9: Track selection and playback initiation via voice detection and gesture.

Drum Beat Creation

Max selects the drum module next. He plays the beats in order by tapping his fingers on the surface, and the system captures his input accurately.



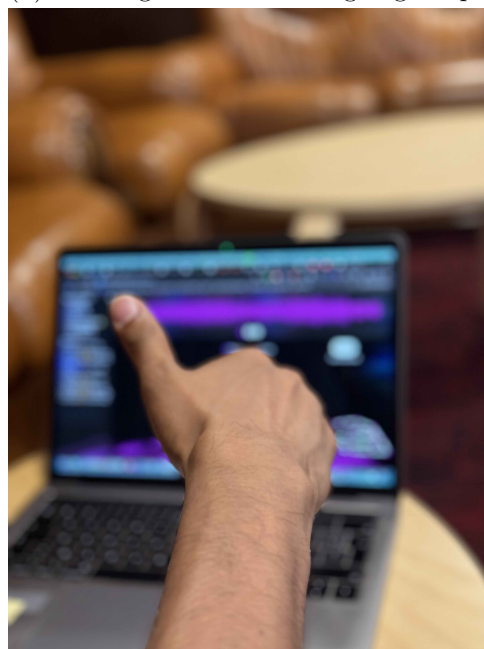
(a) Track selection and playback initiation via voice detection and gesture.



(b) Creating drum beats using finger taps.



(c) Transitioning to the next track by monitoring waveform density.



(d) Adjusting speed and mixing effects using thumb gestures.

Figure 8: Gestures used for various actions in the music creation process.

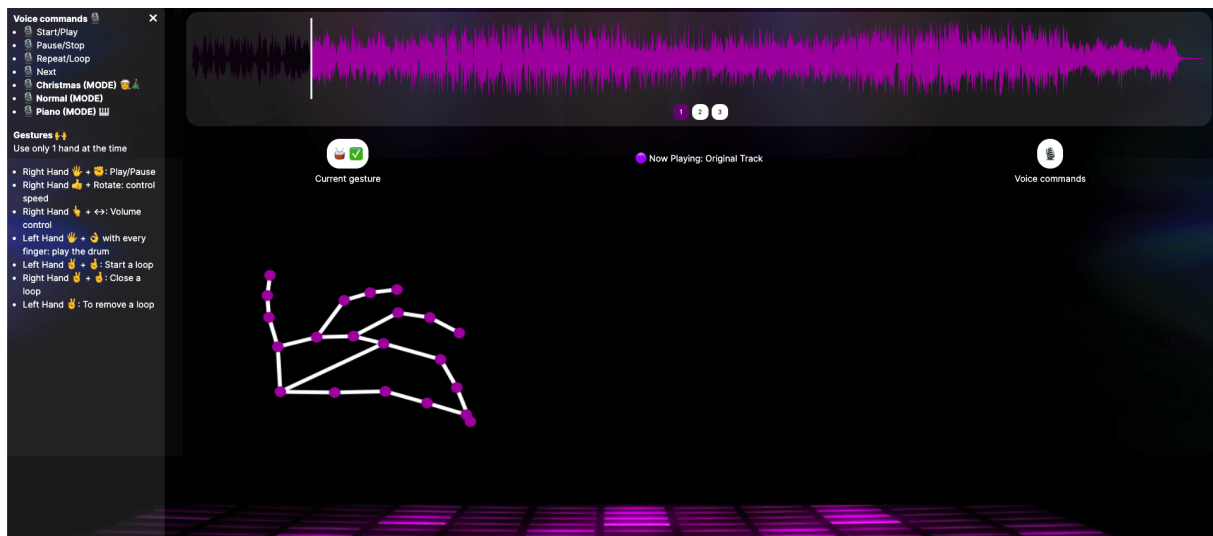


Figure 10: Creating drum beats using finger taps.

Track Transition

After performing the desired beats, Max decides to transition to the next track. He waits for the waveform to show a less dense area, which likely indicates the outro or breakdown where the energy decreases. Max either says the name of the next track to transition or clicks on the next track button.

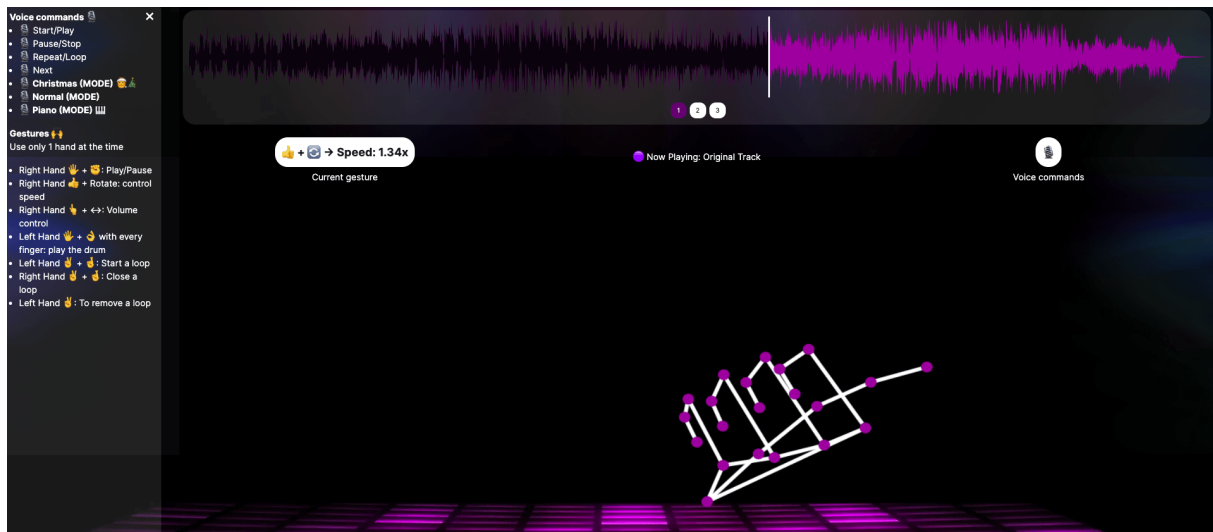


Figure 11: Transitioning to the next track by monitoring waveform density.

Speed Adjustment and Effect Manipulation

Max then decides to change the speed of the music. He rotates his thumb from the center to the right, moving the speed from 1x to 1.5x. As he desires a slow effect, he mixes it by moving his thumb around to create different effects and finally returns to the normal speed.

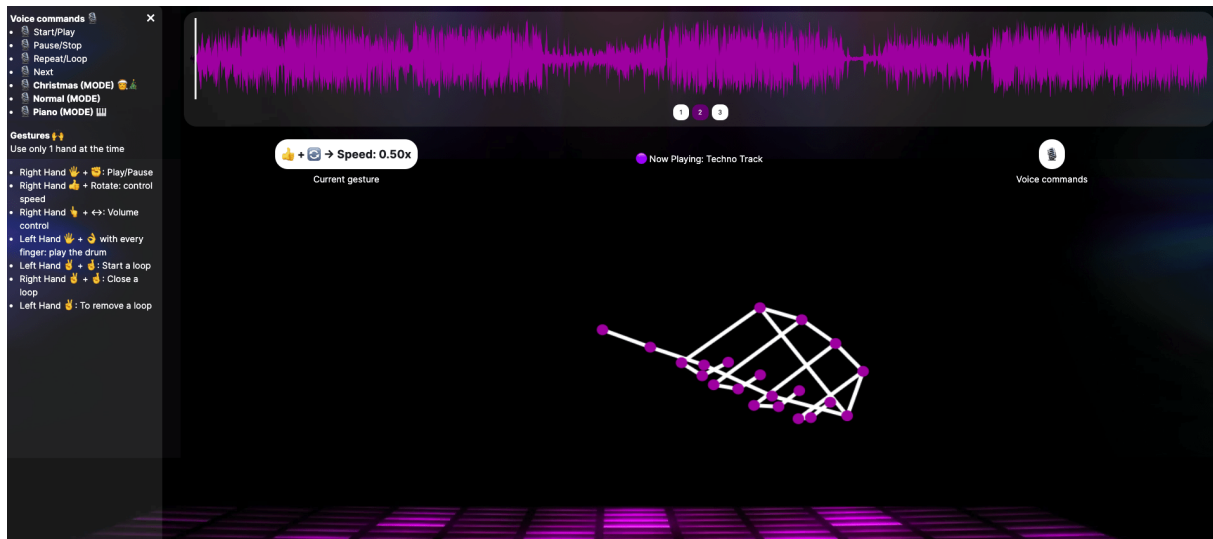


Figure 12: Adjusting speed and mixing effects using thumb gestures.

6.1 User scenario for Self Created Gestures

Further, Max also wishes to experiment with designing his own gesture so he switches to 'create own gestures' in *Tune Crafter*.

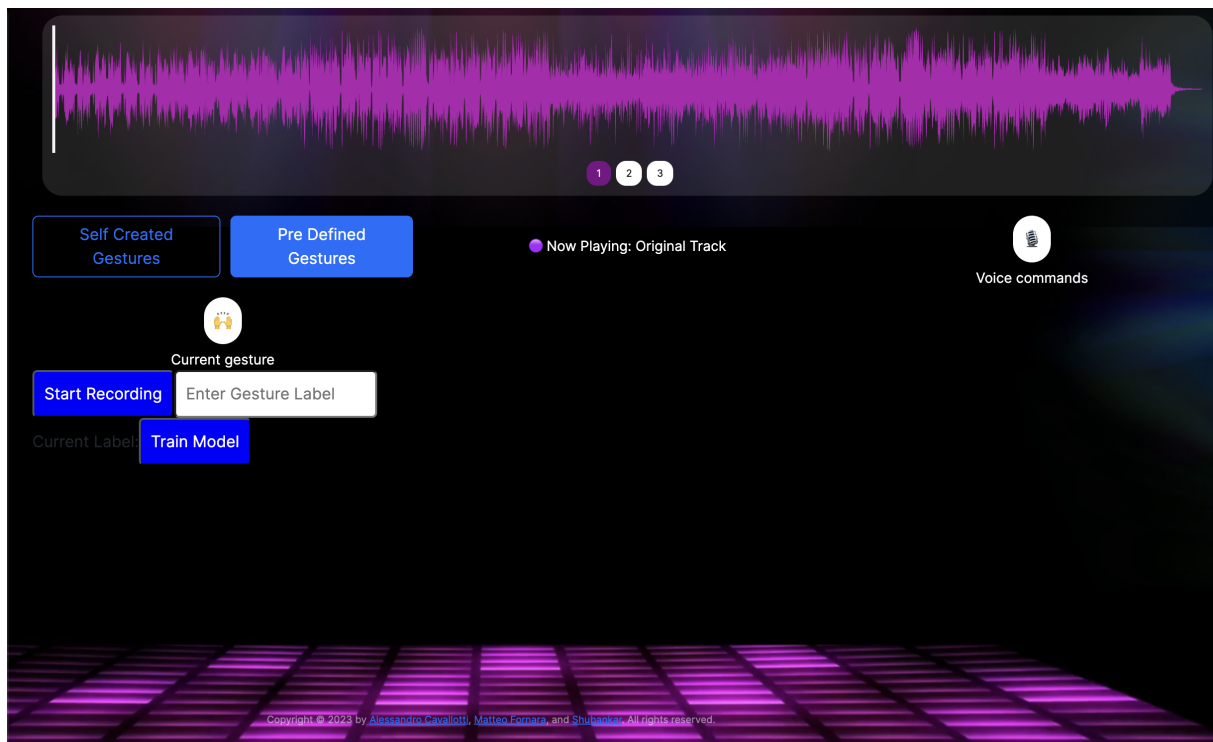


Figure 13: Set the label and start the gesture recording

Creating a Custom Gesture: Max begins by typing the effect he wants to associate with a gesture, such as defining a new volume "up" motion, as the predefined gesture is rather cumbersome. To record the gesture, he clicks the *Record* button and performs the gesture multiple times (e.g., 10–15 repetitions – indication on the bottom of the screen) to provide sufficient training data. This step ensures that the system captures variations in the gesture, improving the model's ability to recognize it accurately, see examples in Figure 13 and 14

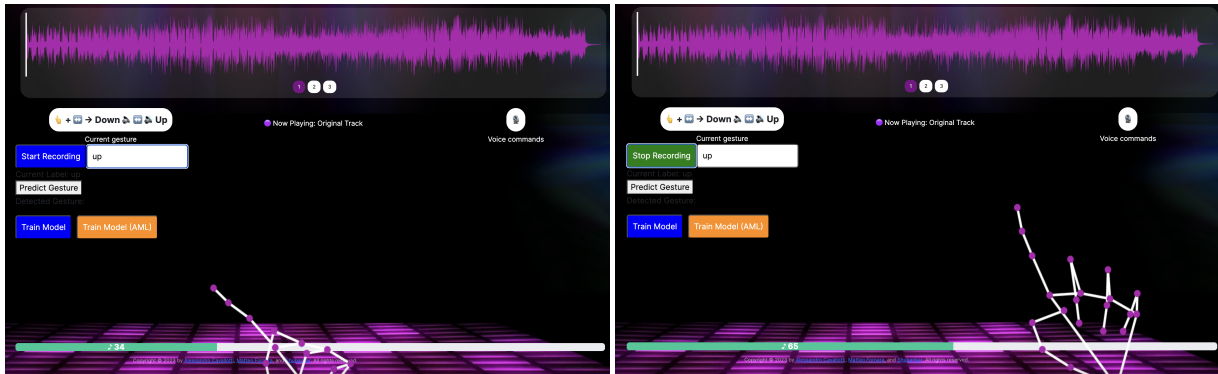


Figure 14: Midway through the gesture

Training the Gesture Model: After recording multiple gestures linked to their corresponding effects, Max initiates the training process by using the buttons below. The system uses the recorded data to train a machine learning model that maps each gesture to its intended effect. This step is crucial for ensuring that the system can accurately interpret Max's gestures during real-time use.

Testing and Validating the Gesture: Once training is complete, Max tests the gesture to see how well the system recognizes it. The interface provides real-time feedback, displaying both the trained gesture label and the model's prediction as Max performs the gesture. This immediate feedback allows him to assess the accuracy of the model. If the prediction quality is low, Max can revisit the training phase to add more data or refine the model, ensuring better performance.

Real-Time Gesture Prediction and Music Effects: Max starts a video or audio playback and activates the real-time prediction mode. As he performs the trained gestures, the system applies the corresponding effects to the music, such as adjusting volume, adding filters, or triggering loops. He observes how the model interprets his gestures and influences the audio in real time, providing a tangible connection between his actions and the music, see Figure 15.

Comparing Models: Similar to D4.8 - Working Prototype and Designs (Revised), also in *Tune Crafter* users can compare different models. So to gain deeper insights, Max runs a comparison between different gesture recognition models. He observes how each model interprets his gestures and applies effects, noting differences in accuracy, responsiveness, and limitations. This comparative analysis helps him understand the strengths and weaknesses of each model, enabling him to choose the one that best suits his needs.

Iterative Learning and Improvement: Through repeated use, Max develops a better understanding of how the models train and perform. He learns to optimize the training process by adding more data, refining gestures, or adjusting parameters. This iterative approach not only improves the system's performance but also deepens Max's familiarity with the technology, empowering him to use it more effectively.

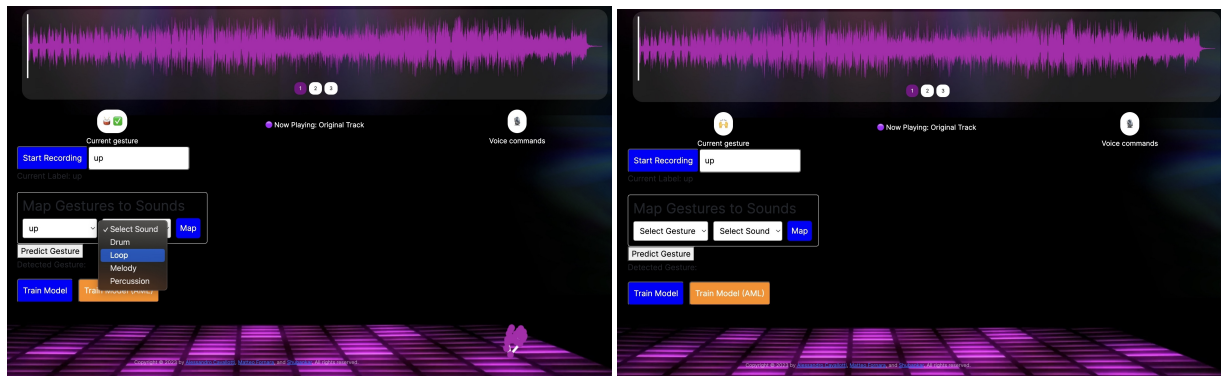


Figure 15: Defined Gesture Overview (left) and Prediction Indicator (right)

7 Discussion

The rise of digital tools in music production has revolutionized the way musicians create and manipulate sound, offering unprecedented opportunities for innovation. However, these advancements come with significant challenges. Many digital music production systems are complex and unintuitive, creating barriers for musicians who rely on spontaneity, fluidity, and real-time interaction in their creative workflows. To address these challenges we developed *Tune Cifter*, a gesture-based system designed to help musicians explore rhythms and music effects in a more intuitive and adaptable way. The project sought to answer three central research questions:

- (1) How do musicians explore and design rhythms currently?
- (2) What limitations do musicians face using digital tools to create rhythms?
- (3) How can systems empower users to create their own tools using AI, while allowing users to stay in control of the creation?

Through a combination of qualitative studies and iterative design processes, we identified key insights into musicians' workflows and needs. Musicians emphasized the importance of layering sounds to build complex compositions, receiving real-time feedback to make precise adjustments, and maintaining granular control over individual audio channels. These insights directly informed the design and development of *Tune Cifter*, a prototype system that integrates gesture-based interaction with explainable AI. *Tune Cifter* offers musicians the flexibility to use *pre-defined gestures* for common sound manipulation tasks, such as adjusting volume, applying filters, or triggering loops, while also allowing them to *define their own custom gestures*. This dual approach ensures that the system can adapt to a wide range of musical styles and personal preferences, bridging the gap between musicians' creative instincts and the capabilities of modern digital tools.

The development of *Tune Cifter* makes three key contributions to the field of digital music production:

1. Empirical Insights: An in-depth empirical study was conducted to identify the challenges musicians face when using digital tools. This study revealed critical pain points, such as the lack of intuitive interfaces and the difficulty of maintaining creative flow, which served as the foundation for the design decisions behind *Tune Cifter*.

2. Gesture-Enhanced Interaction: By applying a design theory methodology, *Tune Cifter* was developed to enhance interaction through gesture control. This approach allows musicians to manipulate rhythms and effects in real time using natural hand gestures, reducing reliance on traditional input methods like keyboards or knobs. The result is a more flexible and creative workflow that aligns with the dynamic nature of music production.

3. AI-Powered Customization: The potential of live gesture detection and music interaction was demonstrated through the integration of **Algebraic Machine Learning (AML)** and other AI models. These technologies enable real-time, explainable interactions, allowing musicians to collect gesture data, train custom models, and apply their unique gestures for sound manipulation. This capability not only enhances the system's adaptability but also empowers users to explore new creative possibilities tailored to their individual needs.

Tune Crafter's ability to combine pre-defined gestures with user-defined customization sets it apart from traditional digital music production tools. Musicians can start with a library of standard gestures for common tasks, ensuring ease of use for beginners, while also having the option to create and train their own gestures for more specialized effects. For example, a musician might define a unique gesture to trigger a specific drum pattern or apply a custom filter, enabling a level of personalization that is rarely found in existing systems. The integration of AI models ensures that these custom gestures are recognized accurately and applied in real time, providing immediate feedback and fostering a more immersive creative experience.

The system's gesture-based interface, combined with AI-powered adaptability, makes digital music production more accessible, intuitive, and responsive to musicians' demands. Compared to D4.8 - Working Prototype and Designs (Revised) and its evaluation in D7.3 - Intelligent Tools for supporting creative professionals (Initial), the gesture definition was expanded from one hand-one point to a five-finger movement-based approach. This allowed *Tune Crafter* to increase the applicability of interactive AML models for creative professionals.

7.1 Limitations

The study had several limitations that influenced its results. To begin, the user research was time-constrained, so only seven participants were recruited, all of whom were men. A more diverse participant pool, including users of different genders, ages, and locations, could have improved the results' generalizability and applicability. Second, although hands-free gestures may allow for easier movement and engagement, they may also cause user fatigue, especially in situations requiring lengthy usage [14]. Furthermore, the system for capturing and detecting live data could benefit from further optimization. This is especially true when both hands are in use, which is a challenge that we hope to address in future versions. Improving the gesture storage and tracking module could increase efficiency. Without these changes, switching between different songs or channels will be difficult.

7.2 Future Work

The future work section for the *Tune Crafter* project outlines several promising avenues for enhancing gesture-based music interaction. Key areas for exploration include expanding gesture libraries to offer a broader range of musical effects, enabling multi-user collaboration for live performances, and integrating with external hardware like MIDI controllers for a hybrid workflow. Additionally, the incorporation of advanced AI models could improve gesture recognition and adapt to user preferences, while developing cross-platform compatibility would increase accessibility for a wider audience. Enhancements to user experience through refined interface design and real-time audio analysis tools could further improve usability and responsiveness. Finally, we aim to evaluate the *Tune Crafter* in the close future and release *Tune Crafter* as an open-source platform to foster community collaboration and innovation.

8 Summary

This Deliverable D7.4 - Intelligent Tools for supporting creative professionals (Final) describes the evolution of our previous work on interactive AML pipeline for creative professionals. By integrating gesture-

based interaction with explainable AI, *Tune Crafter* bridges the gap between human creativity and digital tools, empowering users to explore rhythms and sound effects in a natural and fluid way. The system's dual support for pre-defined and user-defined gestures ensures flexibility, catering to both beginners and advanced users, while its real-time feedback and AML-driven customization enable musicians to train and refine the system to their unique needs.

With that *Tune Crafter* not only addresses the challenges of complexity and rigidity in existing tools but also shows how AML based models could be applied in creative practice. We have successfully produced a working, interactive prototype that allows musicians collect data, train AML, and apply it in a intuitive and complex way than in previous deliverables. Similar to previous deliverables, *Tune Crafter* also provides provides a promising test-bed for exploring how users can use different AI models for real-time predictions in their practice.

This deliverable describes a preliminary study with professional musicians and identifies current understanding and limitations of digital music production. We translated these insights into design implications and the design and implementation of *Tune Crafter*. We demonstrated through a user walkthrough and made the code available to end users so they could experiment with using each model to train different rhythms.

References

- [1] Google Research. “MediaPipe Hands: On-device Real-time Hand Tracking”. <https://google.github.io/mediapipe/solutions/hands.html>.
- [2] Image-Line. “FL Studio”. <https://www.image-line.com/fl-studio/>.
- [3] Ableton. “Live”. <https://www.ableton.com/en/>.
- [4] FabFilter. “Pro-Q 3 Equalizer Plug-in”. <https://www.fabfilter.com/products/pro-q-3-equalizer-plug-in>.
- [5] D. S. P. Valhalla. “Valhalla Vintage Verb”. <https://valhalladsp.com/shop/reverb/valhalla-vintage-verb/>.
- [6] Sidekick Music. “The Art of Sampling in Electronic Music”. <https://sidekick-music.com/2024/05/05/the-art-of-sampling-in-electronic-music/>.
- [7] Piqued. “Getting Started with MIDI Controllers”. <https://piqued.fm/getting-started-with-midi-controllers/>.
- [8] CodeBranch. “Virtual Instruments and the Power of Software for Audio”. <https://www.codebranch.co/post/virtual-instruments-and-the-power-of-software-for-audio>.
- [9] Mogul. “Is it Possible to Learn Mixing and Mastering Without Knowing Music Production?” <https://www.usemogul.com/post/is-it-possible-to-learn-mixing-and-mastering-without-knowing-music-production>.
- [10] YouTube. “EQ Tutorial - How to EQ - 7 Steps to Use Equalizer”. <https://www.youtube.com/watch?v=fLKgk15Ui9s>.
- [11] D. J. Studio. “DJ Mixing for Beginners”. <https://dj.studio/blog/dj-mixing-beginners>.
- [12] Mark Weiser. “The Computer for the 21st Century”. In: *Scientific American* 265.3 (1991), pp. 94–104.
- [13] Hiroshi Ishii and Brygg Ullmer. “Tangible bits: towards seamless interfaces between people, bits and atoms”. In: *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. ACM, Mar. 1997. DOI: 10.1145/258549.258715.
- [14] Mathieu Nancel et al. “Mid-air Pan-and-Zoom on Wall-sized Displays”. In: *Proceedings of the CHI 2011 Conference on Human Factors in Computing Systems, Vancouver, Canada*. May 2011, pp. 177–186.
- [15] Theophanis Tsandilas and Wendy E. Mackay. “Knotty Gestures: Subtle Traces to Support Interactive Use of Paper”. In: *Proceedings of the International Conference on Advanced Visual Interfaces. Proceedings of the Working Conference on Advanced Visual Interfaces, ACM*. ACM, 2010, pp. 147–154.
- [16] David Wessel and Matthew Wright. “Problems and Prospects for Intimate Musical Control of Computers”. In: *Journal of New Music Research* (2001).
- [17] Wendy E. Mackay. “Responding to Cognitive Overload: Co-adaptation between Users and Technology”. In: *Intellectica* 30.1 (2000), pp. 177–193.
- [18] Xiaohan Peng, Janin Koch, and Wendy E. Mackay. “DesignPrompt: Using Multimodal Interaction for Design Exploration with Generative AI”. In: *Designing Interactive Systems Conference*. ACM, July 2023, pp. 804–818.
- [19] Tina Blaine and Sidney Fels. “Contexts of Collaborative Musical Experiences”. In: *Proceedings of the Conference on New Interfaces for Musical Expression, 2003*. 2003.
- [20] Martín Abadi, Ashish Agarwal, Paul Barham, et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. <https://tensorflow.org>. Software.
- [21] Baptiste Caramiaux. *Machine Learning in Interaction: Tool, Material, Culture*. Université Paris-Saclay: Human-Computer Interaction [cs. HC], 2023.
- [22] Sun Jiao et al. “Investigating Explainability of Generative AI for Code through Scenario-Based Design”. In: *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, 2021*. 2021.

- [23] Rebecca A. Fiebrink and Baptiste Caramiaux. *The Machine Learning Algorithm as Creative Musical Tool*. Ed. by Roger T. Dean and Alex McLean. Oxford University Press, 2018. DOI: 10.1093/oxfordhb/9780190226992.013.23. URL: %5Curl%7Bhttp://dx.doi.org/10.1093/oxfordhb/9780190226992.013.23%7D.
- [24] Nick Bryan-Kinns et al. “Exploring XAI for the Arts: Explaining Latent Space in Generative Music”. <https://arxiv.org/abs/2308.05496>. 2023. arXiv: 2308.05496.
- [25] George Sung et al. “On-device Real-time Hand Gesture Recognition”. <https://arxiv.org/abs/2111.00038>. 2021. arXiv: 2111.00038.
- [26] Jeffrey D. Scargle et al. “Studies in astronomical time series analysis. VI. Bayesian Block representations”. In: *The Astrophysical Journal* 764.2 (2013), p. 167.