A Lack of Data for Transcribing Audio into Musical Notation

- An ongoing task of great interest for both the fields of AI and music information retrieval is the ability to automatically transcribe audio into musical notation.
- On one side, most musicians record their performances as raw audio through a microphone. While easy to play back, the notes and details cannot be discreetly read by a computer, or directly converted into sheet music. Notation based audio formats exist, mainly MIDI, but the issue then becomes how to transcribe the raw audio into this discrete, MIDI format.
- Our project aims to be a conclusive first step in solving this difficult task. The MIDI Performance Data Website is a platform where the collection of audio-MIDI synchronized data can be crowdsourced to the musician volunteers, and a place where the data needed for the main task can be openly searched for and acquired.
- Users can connect any MIDI-enabled digital instrument to their computer and navigate to the website, where they can record musical performances to contribute to the dataset. The platform records both the raw acoustic signal of their musical performance, as well as the discrete MIDI values being made by the instrument.
- With a large enough dataset, users can then search, filter, and download data of synchronized audio and MIDI files, which can be used to effectively train machine learning models to transcribe raw audio recordings into symbolic MIDI data.

Oregon State University

MIDI Performance Data

An open source solution for recording and accessing audio and symbolic music data, all on one website.

Recording Process

Recording is performed with the `WebMidi.js` library, which interfaces with the browser's WebMidi API. It identifies all MIDI devices connected to the computer and facilitates interfacing with the devices through the browser. Messages sent from connected devices are recording and stored to later be serialized into a usable MIDI file. Serialization is handled through a separate library `jzz-midi-smf`. Audio is also recorded using a selected microphone. The collected data is then labeled by the user, with metadata such as title, composer, and instruments used, and the data is stored in the database.

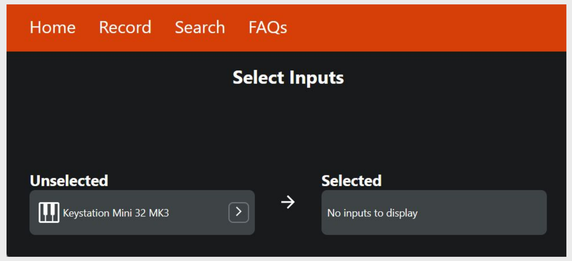

Figure 1: Selecting recording devices.

Search Process

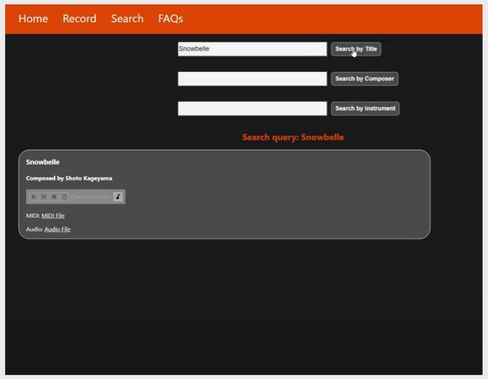

Figure 2: Search query by song title.

Search allows the user to query and interact with the data repository in multiple different ways. An entry in the dataset consists of a pair of raw audio and MIDI files submitted by a user's recording, along with metadata such as title, composer, and instruments used. In the search, a user is able to filter and select data using any of these three accompanying details, title, composer, and instruments. Once the filtered data collection has been fetched and loaded, a user can playback the audio of any individual entry, as well as download singular entries. The entire query itself can also be downloaded as a whole to be used for the main intended purpose: training machine learning models to convert between the two audio data types.

Backend and Database

To host our database, we selected Google's FireBase as our cloud storage solution because of the ease of integration into our React-based pipeline to interface with our web app. We connect the two with Javascript and maintain a database with strong documentation. All data are stored as "blobs" (a file-like object used in JavaScript) as key values in our database. This organizational system support intuitive queries to quickly locate and return the desired files and other information.

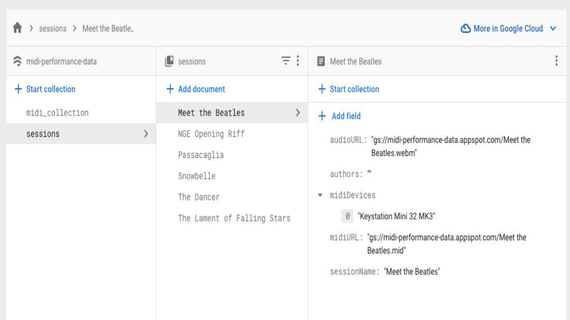

Figure 3: Firebase database layout

About Our Team

Scot Rein
- Search and Statistics

Mason Mann
- Frontend and Audio/MIDI recording

Caleb Knight
- Backend Developer and Firebase Coordinator

Patrick Donnelly
- Project Partner, Computer Science

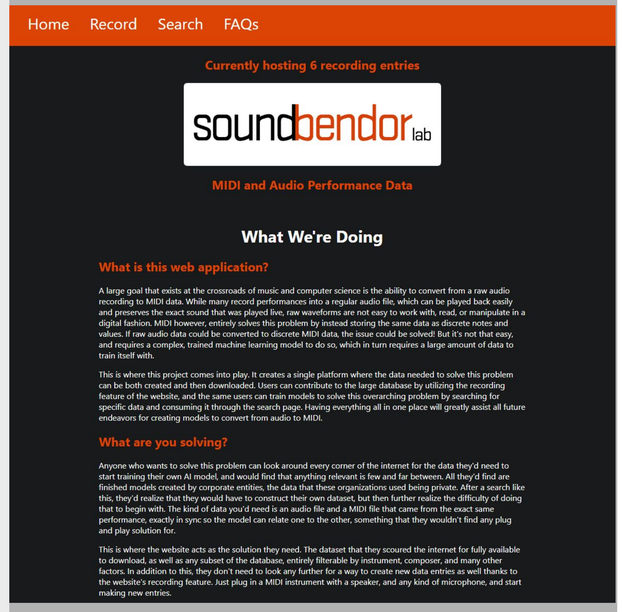

Figure 4: Homepage of MIDI Performance Data project.

soundbendor lab