Symphony AI

Team D5: Eli Wirth-Apley, Vedant Bhasin, Sachit Goyal
18-500 Capstone Design, Spring 2023
Electrical and Computer Engineering Department
Carnegie Mellon University

Product Pitch

Introducing our innovative system that uses generative AI models to generate backing tracks for guitarists! Whether you're a seasoned professional or just starting out, our system offers a unique and exciting way to enhance your musical creativity.

With our system, you have the option to **either upload a WAV file to condition the model** or **generate music unconditionally.** This means that you can create custom backing tracks that perfectly match the style and feel of your original music, or simply let the AI model generate entirely new and unique tracks for you to jam along with **in different genres.**

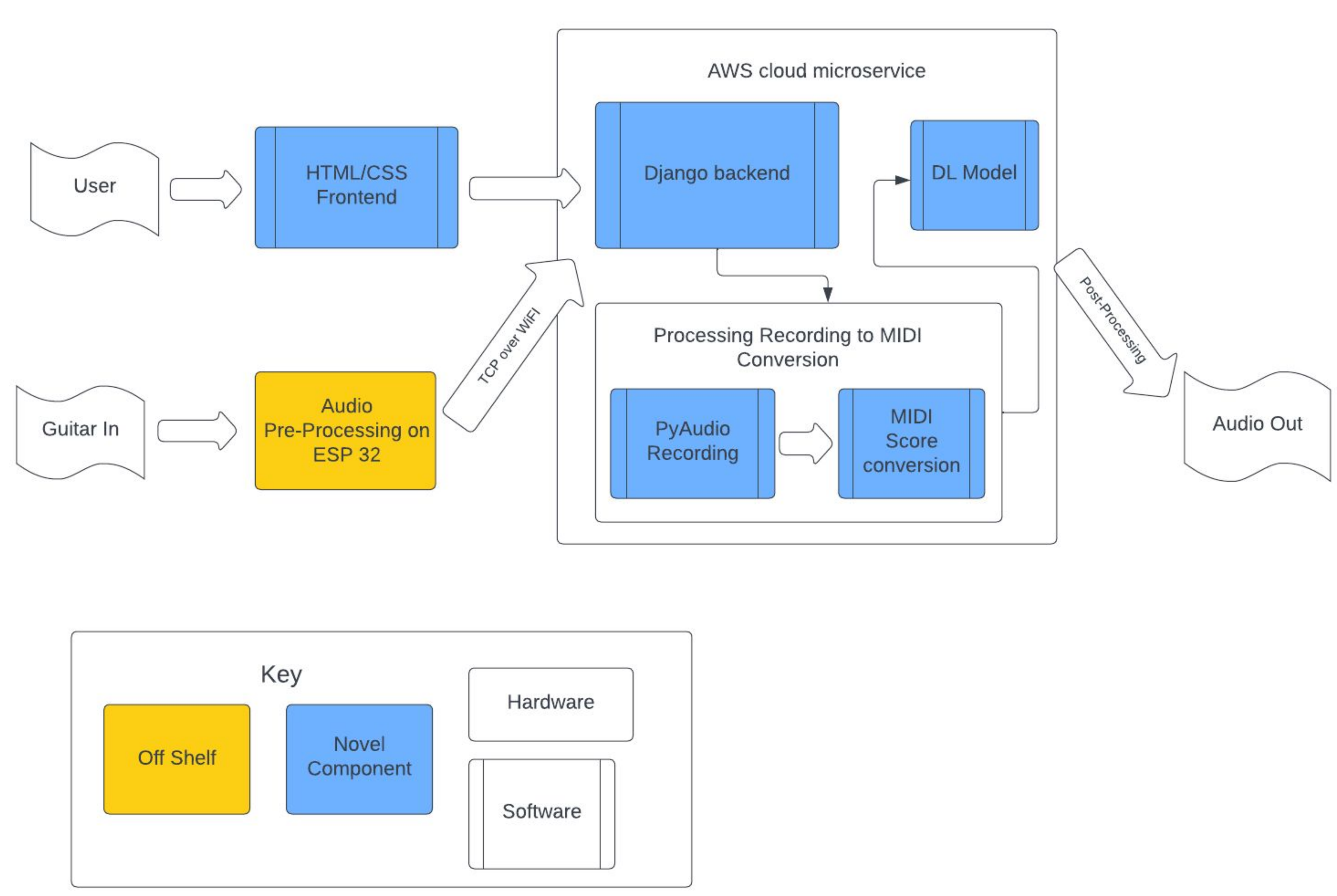
Our system is perfect for a wide range of applications, from practicing and rehearsing to recording and performing. Whether you're looking to improve your skills as a guitarist, collaborate with other musicians, or simply have fun experimenting with new sounds and styles, our system offers endless possibilities.

Some specific use cases for our system include:

- **Practicing**: Our system can generate backing tracks to help you practice specific techniques, such as scales, chord progressions, or improvisation. You can adjust the tempo, key, and other parameters to suit your needs and preferences.
- **Recording**: Our system can generate custom backing tracks for your original music, giving you the freedom to experiment with different styles and arrangements. You can easily export the tracks as WAV or MP3 files to use in your recordings.

Overall, our system is a must-have tool for any guitarist looking to take their music to the next level. With its innovative use of generative AI models and its wide range of applications and use cases, our system offers endless possibilities for musical creativity and expression.

System Architecture



Conclusions & Additional Information

In conclusion, our project taught us valuable lessons in goal setting and implementation. While it is tempting to aim for the stars and replicate state-of-the-art results from research papers, it is crucial to set realistic expectations and account for potential implementation challenges. Despite facing some limitations in functionality compared to our initial aspirations, we were able to achieve the core objective of our project.

Moving forward, we believe that there are numerous exciting possibilities for extending this project. For instance, training the generative model on larger datasets could significantly enhance the quality and diversity of the generated music. Another intriguing challenge would be to create a system that continuously generates music and seamlessly plays along with the user, a task that would require careful engineering given the fixed size and shape requirements of deep learning models. Ultimately, we are proud of what we have accomplished with this project and are excited to see where it can lead in the future.

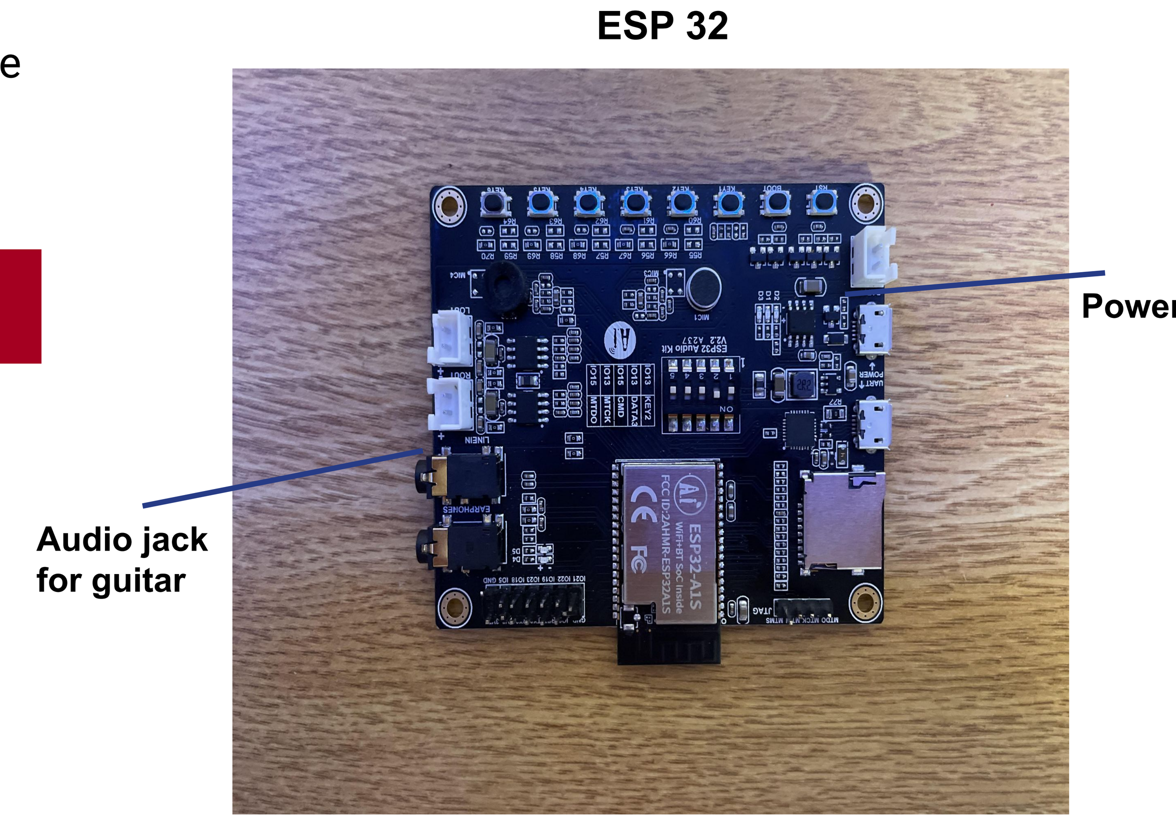
http://course.ece.cmu.edu/~ece500/projects/s23-teamd5/

System Description

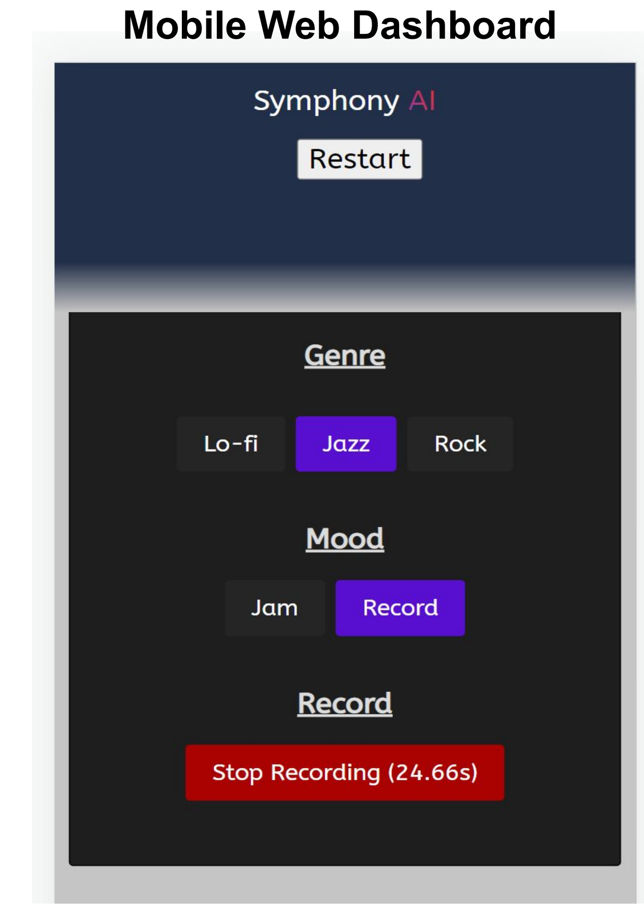
Our system consists of three primary subsystems:

1. **Guitar recording using ESP32**
   This subsystem deals with recording raw audio from the ESP 32 board with the guitar connected to the board and transmitting the data to the backend over Wifi using TCP communication to the Django backend.
2. **Audio to MIDI conversion**
   In Python using various libraries including SciPy, Aubio and MIDIUtil pitch detection is performed via the Yin Autocorrelation method. Using note onset times and normalized loudness, a midi file is generated representing the transcribed monophonic clean tone guitar input wav file.
3. **Attention based music generation**
   This subsystem takes in the transcribed MIDI file from the previous subsystem and performs conditioned music generation using a transformer based architecture deep learning architecture.
4. **Web application interface on Mobile**
   This is how the user interacts with our project. The user has the option to record audio to generate ML-generated accompaniment, or they can choose 'jam' mode where the transformer generates music from it's own learned latent space. The user can control the guitar recording, genre and download the generated music through the frontend.

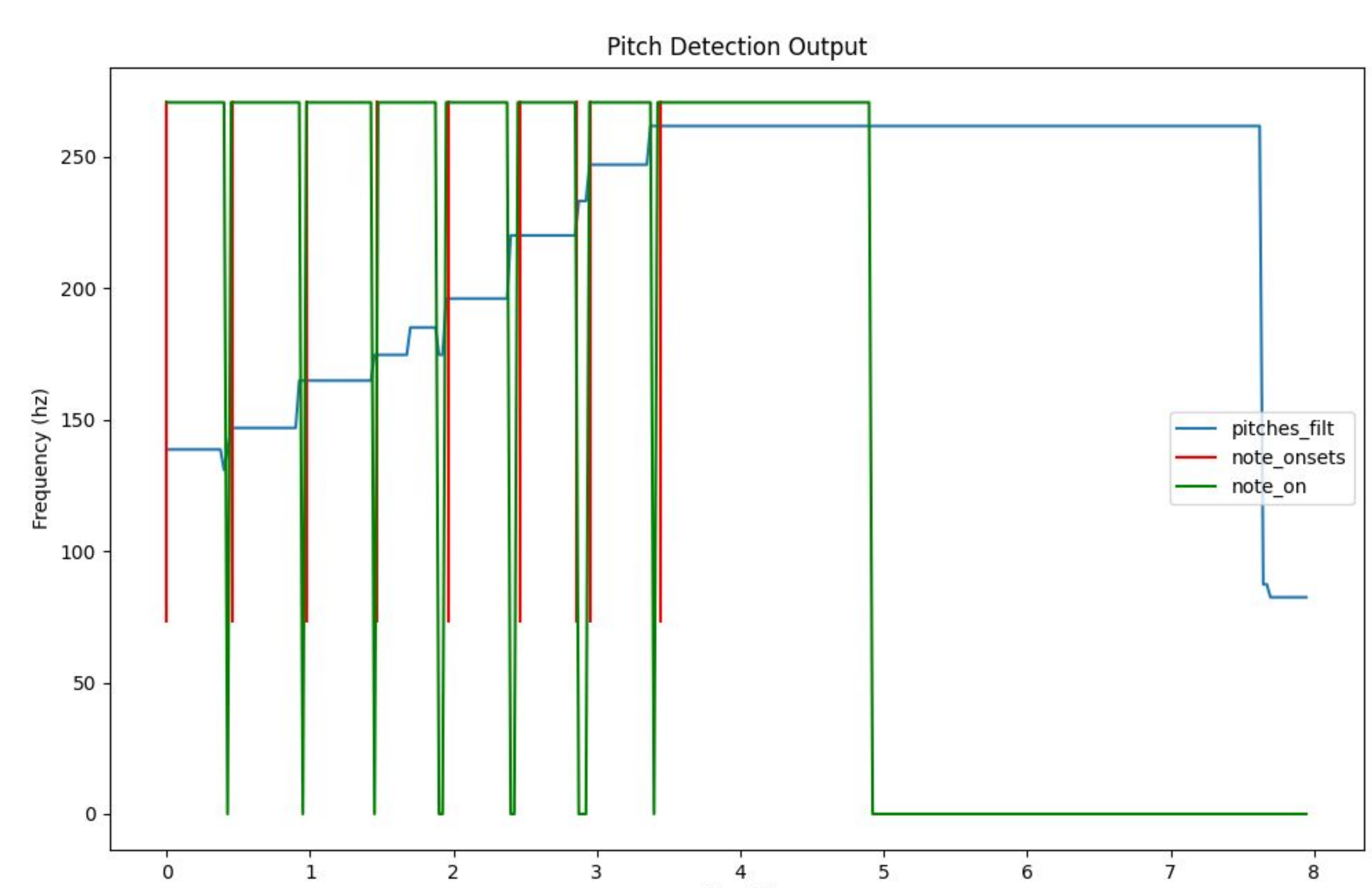
ESP 32



Mobile Web Dashboard



System Evaluation



The above figure shows the results of our pitch detection script when run on a monophonic guitar c major scale wav file. Our pitch detection correctly identifies all notes in the scale. It incorrectly includes one extra note. Across multiple tests on similar inputs, our pitch detection performs similarly. A full system test involves generating ~30 wav files from the esp32 and analyzing the output of the pitch detection by hand. according to known inputs. At present, we sacrifice computation time for accuracy.