Music Track Similarity

Dhawal Modi[*]
Sravan Jayati[*]
dmodi2@ucmerced.edu
sjayati@ucmerced.edu
University of California, Merced
Merced, California, USA

ACM Reference Format:
Dhawal Modi and Sravan Jayati. 2023. Music Track Similarity. In *Proceedings of EECS257 Project Final Report*. University of California, Merced, 3 pages.

1 INTRODUCTION

The problem statement of this project is to find the closest match for a given audio recording of someone playing a classical music track using any single instrument, from a list of classical music tracks in our dataset using discrete time signal processing concepts.

We compare different audio samples generated for different instruments (Violin, Piano, Guitar etc) using MIDI files and, find a measure of similarity between the samples and the original audio track. We use the MIDI files provided in the MusicNet database [3][2] to generate our test audio files.

Audio similarity measurement is an important research area in the field of audio signal processing. It is used to quantify the degree of similarity between two or more audio signals, which can be useful in various applications such as music recommendation systems, audio search engines, and content-based music retrieval systems.

For instance, in music recommendation systems, audio similarity measures can be used to extract underlying sound characteristics from audio data and calculate the similarity between different music tracks. This information can then be used to construct a data structure describing user preferences and recommend similar music tracks based on the similarity of user models.

2 SIGNAL PROCESSING GOAL

In this project, we use concepts such as Resampling, noise reduction using Spectral Gating, STFT, Spectrogram, Chromagram, Dynamic Time Warping (DTW), and get a similarity score which would finally be used to find the closest match from the dataset.

Applications like Shazam create audio fingerprints for each song based on relative positioning of spectral peaks in a spectrogram and then find the closest match using these audio fingerprints. However, this approach does not account well for different variations of the same song, i.e. covers of a certain song would only be matched with the same exact cover but not the original song. This approach is sensitive to relative tempo differences between the query and database.

We aim to address this issue with the use of chromagrams, which are designed better for musical notation based audio matching.

3 DATASET

The MusicNet dataset[3][2] contains 330 audio recordings of orchestral pieces which may or may not contain multiple instruments in the recording. For each audio recording, the dataset also contains a corresponding MIDI file, which is a digital representation of all the musical notes and their properties (such as timestamp, loudness, etc).

We use the MIDI files to generate audio files of the same orchestral pieces using a single instrument and, we consider this as our test dataset. We also create a version of this dataset by artificially adding White Gaussian Noise with varying SNR values to see our algorithm's efficacy with noise.

4 SIGNAL PROCESSING METHODS

Obtaining the similarity of two audio samples is carried out in five steps/stages:

4.1 Storing resampled chromagrams of the original data

Initially, we perform a process of resampling and retrieval of chromagrams from the original dataset files. These chromagrams are stored for the purpose of comparing a given test audio file and determining their degree of similarity. More on resampling and chromagrams can be found in the subsequent sections.

4.2 Resampling

We resample our audio files to 5kHz. Audio files typically are of 44.1kHz. This is because the human ear's perceivable frequency range is 20Hz to 20kHz, and to maintain the Nyquist sampling rate for audio playback as well as accounting for anti aliasing filter, the chosen standard for audio files has been 44.1kHz. However, if audio playback is not important and we are only interested in audio analysis, 22.5kHz is a commonly used sampling rate. Further, because we are working with orchestral pieces and the range of frequencies for most musical instruments go up to 4.2kHz, resampling our audio files to 5kHz provides us with faster inference time while preserving the important frequency components.

4.3 Noise reduction using Spectral Gating

We tried to refrain from using Butterworth or Chebyshev filters because these act as cutoff filters, cutting off a range of frequencies that we might not need. However, if White Gaussian noise with a low SNR is present in an audio file, cutting off a frequency range doesn't improve the quality of the signal within the frequency range that we are interested in. Instead, we use Spectral Gating.

[*] Both authors contributed equally to this research.

EECS257 Project Final Report, December 5 2023,

Spectral gating is used in audio and music analysis to control or manipulate the dynamics of a sound based on its frequency content. It involves dividing an audio signal into different frequency bands and then applying gating or dynamic processing independently to each band. The primary goal of spectral gating is often to enhance or modify specific frequency components of a sound while leaving others unaffected.

4.4 Chromagram analysis

Although spectrograms are great way to analyze audio signals, we cannot just rely on a spectrogram due to a few challenges. A certain classical track played using different instruments would result in different degrees of harmonic frequencies in the spectrogram.

To account for such challenges, we convert the spectrogram to a chromagram[1], which is essentially a quantized version of the spectrogram, binned into 12 bins that correspond to the 12 musical notes (i.e. Do, Re, Mi,..). Hence, regardless of the type of instrument played, we obtain a similar representation to the file from our dataset as long as the musical notes played remain the same.

4.5 Dynamic Time Warping (DTW)

To get a similarity score between the chromagrams of a test file and a file from the dataset, we use Dynamic Time Warping (DTW). DTW is an algorithm that can be used to measure the similarity between two temporal sequences. In the context of chromagrams, DTW can be used to compute the distance between two chromagrams by finding the optimal alignment and cost matrix path. Another advantage of using DTW is that it allows for the comparison of musical pieces with varying tempo or timing, making it valuable for tasks like music similarity.

Using the DTW, we find the least cost path and then compute the average of all the cost values along the least path. Finally, we use the average cost value to find the closest match among all the files from the original dataset.

5 SUCCESS METRICS

For our success metric, we primarily use the accuracy of the matches from test files to their corresponding original audio files from the database.

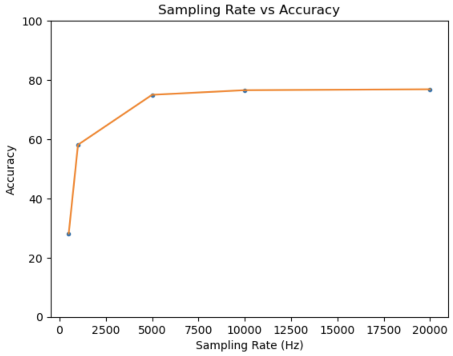
Accuracy calculation: For each test file, we calculate the average cost between the test file and all of the 330 files in the original dataset. Considering the file with least average cost value as a match, we computed accuracy as the percentage of files that were correctly matched with their original file in the dataset.

Noise reduction analysis: Additionally, we perform an analysis on 4 test sets: without noise, with noise of SNR 1, with noise of SNR 5 and noise of SNR 15. We address the drop in accuracies with the addition of noise. We use the accuracy achieved on the test set without noise as a baseline, perform noise reduction using Spectral Gating on test sets of varying SNR values and, finally show an improvement in accuracies with noise reduction.

6 RESULTS

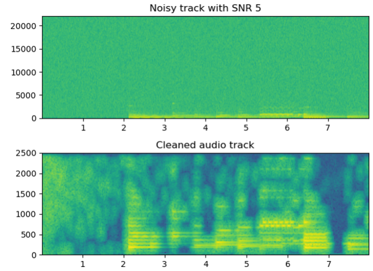
6.1 Resampling effect on accuracy

We observe that the accuracy has little to no impact for sampling rates higher that 5000Hz and, there is a significant drop in accuracy below 5000Hz. This is expected as important frequencies related to the orchestral pieces are discarded below 5000Hz.



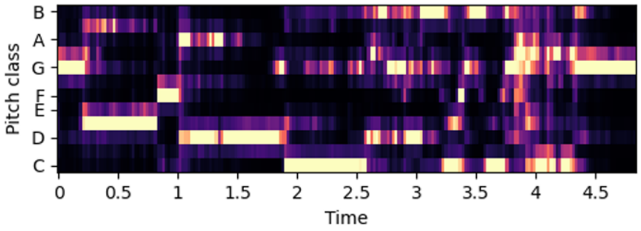
6.2 Noise reduction with Spectral Gating

With a value of SNR 5, we observe a lot of random, unwanted noise beyond 5000Hz. After performing noise reduction using Spectral gating, frequencies belonging to the musical notes in the audio track are isolated and the remaining unwanted noise seems to be removed.

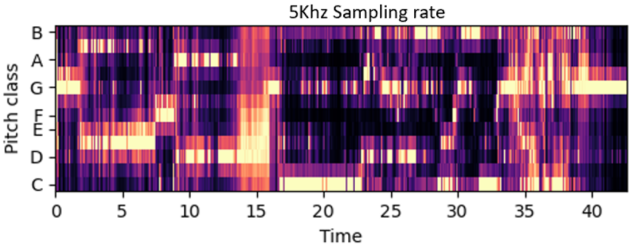


6.3 Resampling effect on Chromagrams

We observe that all the important frequencies corresponding to the musical notes in the audio track are well preserved when using a sampling rate of 5kHz as well. From this, we confirm that using a sampling rate of 5kHz would have no impact in a music audio matching application.



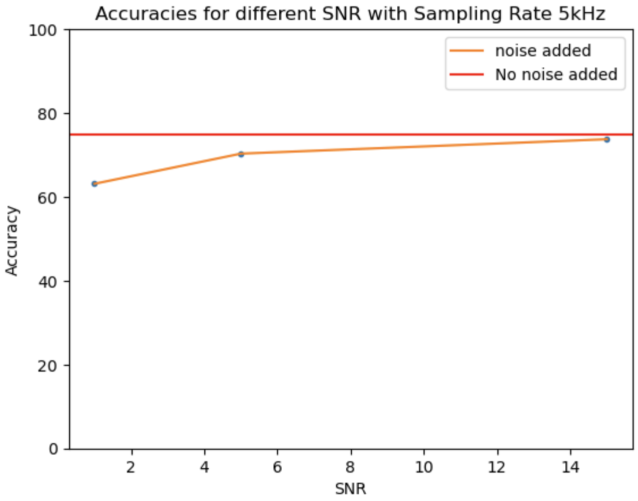
5Khz Sampling rate



44Khz Sampling rate

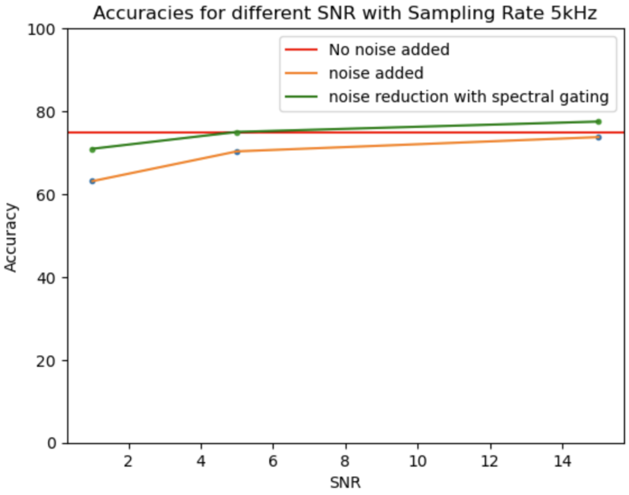
6.4 Accuracy on test set with varying SNR

As expected, we see that with a lower value of SNR, there is a higher drop in accuracy compared to the test set excluding any noise.



6.5 Accuracy on test set with varying SNR after noise reduction

After performing noise reduction using Spectral Gating on the test sets with noise, we observe a significant improvement in accuracy. We also find that for an SNR value of 15, there is an improvement over the accuracy achieved using the test set without any added noise. We believe this is because the generated test set could also contain a low amount of noise that could have a minor impact on accuracy.



7 CONCLUSION AND FUTURE DIRECTION

In our work, we successfully designed a Signal Processing based algorithm to retrieve the closest match from our database of 330 audio recordings of orchestral pieces for any given audio file. The highest accuracy we managed to achieve was 77.5% from our test dataset with an SNR value of 15. We also verify the tempo (speed of a song) invariance effect of using DTW, as our test dataset contained files with a different tempo when compared to their corresponding files from the original dataset.

Although we deliberately avoided using machine learning algorithms in order to avoid black boxes in our system and to emphasize Signal Processing concepts in our work, machine learning algorithms can be explored to improve the accuracy further. Also, the database could be extended to files which don't just contain orchestral pieces.

8 TEAM INFORMATION

Sravan created the custom dataset from the MusicNet database and worked on chromagram analysis, experimentation and testing. Dhawal created the program for comparing DTW implementations and worked on Spectral Gating and the system design.

REFERENCES

[1] Meinard Müller, Frank Kurth, and Michael Clausen. 2005. Chroma-Based Statistical Audio Features for Audio Matching. In *Proceedings of the Workshop on Applications of Signal Processing (WASPAA)*. New Paltz, New York, USA, 275–278.

[2] John Thickstun, Zaid Harchaoui, Dean P. Foster, and Sham M. Kakade. 2018. Invariances and Data Augmentation for Supervised Music Transcription. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[3] John Thickstun, Zaid Harchaoui, and Sham M. Kakade. 2017. Learning Features of Music from Scratch. In *International Conference on Learning Representations (ICLR)*.