Music Identification through Audio Fingerprinting

Group 1

Yashraj Kakkad - *AU1841036*
Prayag Savsani - *AU1841035*
Yashil Depani - *AU1841005*
Harshil Mehta - *AU1841010*

Guided by - Prof. Ashok Ranade

The group photo goes here

I. Objective

The objective of the project is to recognize a song from an arbitrary 30 second sample through appropriate audio fingerprinting techniques. To accomplish that, a number of songs are broken into chunks. The best frequencies across a spectrum are taken and they are hashed in an appropriate data structure. The obtained input from the user undergoes a similar procedure and is then compared with the hash values of the song. The song which matches with the input the most is displayed as a result to the user.

This was a very brief outline. The coming sections exactly explain what we have done.

II. Sampling

Music is typically sampled at 44.1 kHz. This is because of a theorem by Nyquist and Shannon which requires $f_d >= 2f_s$ where f_d is the digital frequency and f_s is the sampling rate. The number, in practice, is greater than 2. Maximum sound frequency, as we know, is 20 kHz.

We'll be using Fast Fourier Transform at a later stage to obtain the frequency spectrum of several chunks of the song. Running it on even a few hundred songs would take days. Therefore, we downsample the recorded sample by a factor of 4. As a consequence, the maximum sound frequency in our audio sample comes down to 5kHz. This is not a problem as the frequency spectrum of most of the song lies below 5kHz. The gains are therefore much higher than the losses.

III. Filtering

Aliasing refers to the distortion that results when a signal reconstructed from samples is different from the original continuous signal. We filter the high-end frequences before downsampling to avoid this phenomenon.

IV. Fast Fourier Transform

Discrete Fourier Transform converts a signal from the time domain to the frequency domain. The formula is:

$$X(n) = \sum_{k=0}^{N-1} x[k]e^{-j(2\pi kn/N)}$$

Fast Fourier Transform is an algorithm that computes Discrete Fourier Transform. As the name says, it is "faster" than actual Discrete Fourier Transform. To make a comparison, Discrete Fourier Transform requires $\mathcal{O}(N^2)$ while its counterpart only requires $\mathcal{O}(NlogN)$, where N is the number of samples. This is a huge improvement since N is typically quite large in our case.

Speed is important in this case as the quicker we get the answer, the better experience we provide to the user. Also, the database of songs to be maintained is large.

V. Section Title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

VI. Section Title

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.