IEEE NITK CHAPTER

PROJECT – Audio Fingerprinting with Python

|  |  |
| --- | --- |
|  | **ABSTRACT**  *An audio fingerprint is a condensed digital summary generated from an audio signal that can be used to identify an audio sample or locate similar items in an audio database. Apps like Shazam, Sound Hound, etc., are able to recognize a song from a large cluster of audios. This can’t be done just by using brute force to compare an audio sample to every song in the database. In this project we will use the hashing method to explore audio fingerprinting. We will also be working on a search algorithm to identify the audio fingerprint of the given audio sample from a database of songs.* |

# Section 1 – project overview

|  |  |
| --- | --- |
|  | *The proposal is divided into two sections namely – Project Overview and Literature Review. The first section will cover the motivation, objectives, methodology, timeline, budget and deliverables of the project. The second section contains the literature review regarding the different phases of the project.* |

## The Motivation

The very idea of how a smartphone can recognize a song from a vast library of audio through the Shazam app is incredible. Especially since this recognition happens after only 5 short seconds of listening to the unknown audio sample.

## The Opportunity

Through this project, we can explore and learn the working of many fields of interest such as signal processing, hash functions, etc.

Audio fingerprinting itself is an up-and-coming field with numerous applications. Audio resources have become an important source of information, but these resources require to be handled appropriately in order to be useful.

The Shazam app, from which this project seeks inspiration from, is not merely a search-and-identification tool. From isolating the required sample of the unknown song from a noisy environment, to processing it in a compact way, to efficiently storing it in a database, or ensuring quality of identification despite having the audio input compressed, cropped, etc., each of these by themselves can be researched on for suitable applications.

The greatest opportunity this project offers is that this is simply the starting point for those interested in this field.

## The Objective

This project aims to create a program that can identify which song a short audio sample originated from.

## Project Approach

TBD

## Project Deliverables

The following are the full list of deliverables which will be achieved through the entire project timeline:

|  |  |
| --- | --- |
| Deliverable | Description |
| Spectrograms | Realization of music as a signal.  Creating a spectrogram of an audio file. |
| Fingerprinting a song | Combining peaks of the spectrograms into fingerprints using a hash function. The hash represents a unique fingerprint of a song. |
| MySQL | Making a MySQL database. Storing and retrieving information from a database. |

## Timeline for Execution

Key project dates are outlined below. Dates are best-guess estimates and are subject to change until a contract is executed.

|  |  |  |  |
| --- | --- | --- | --- |
| Description | Start Date | End Date | Duration |
| Project Phase 1 | | | |
| Milestone 1:  Learning the basics of Python, getting familiar with the Jupyter notebook, OpenCV, and NumPy |  |  | 2-3 Weeks |
| Milestone 2:  Making Spectrogram of a given audio file, and finding peaks in the in it. |  |  | 1 month |
| Project Phase 2 | | | |
| Milestone 3:  Learning the basics of MySQL, making a database, adding and retrieving information from a database. |  |  | 1-2 weeks |
| Milestone 4:  Learning and implementing Fingerprint Hashing. Implementing learning of a song. |  |  | 1 month |
| Project Phase 3 | | | |
| Milestone 5:  Implementing Fingerprint Alignment |  |  | 1 month |
| Milestone 6:  Testing the model, from taking the test audio sample from various sources, and analyzing it to find the minimum time of the sample for optimized searching. |  |  | 1 Week |