End Semester Presentation
Machine Learning & Pattern Recognition

Humming Based Song Identification

–Utkarsh, Shruti & Arya

Problem Statement

People often find themselves in situations where they can recall only fragments of a tune but cannot pinpoint the exact song or artist.

This problem is particularly prevalent when lyrics or complete audio are not accessible, leaving individuals with no means to satisfy their curiosity or retrieve the desired musical content.

Earworms can be super annoying, for you...
and people around you!

Tune stuck in your head?

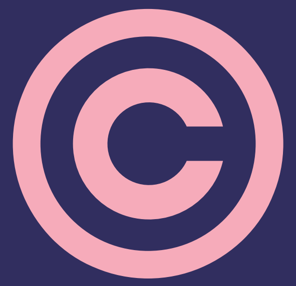
Proposal & Impact

- A novel ML Model capable of recognizing & identifying songs from hum and whistle inputs
- Transform user experience, empowering users to explore & engage with their favourite tunes
- Enhancing music discovery, accessibility, & engagement, while also contributing to technological advancements in the field of audio signal processing & ML.

Potential Applications

- Integration with streaming platforms like Spotify
- Additional feature in voice assistants
- Standalone mobile app

- Intelligent tutoring systems for music education
- Similarity score

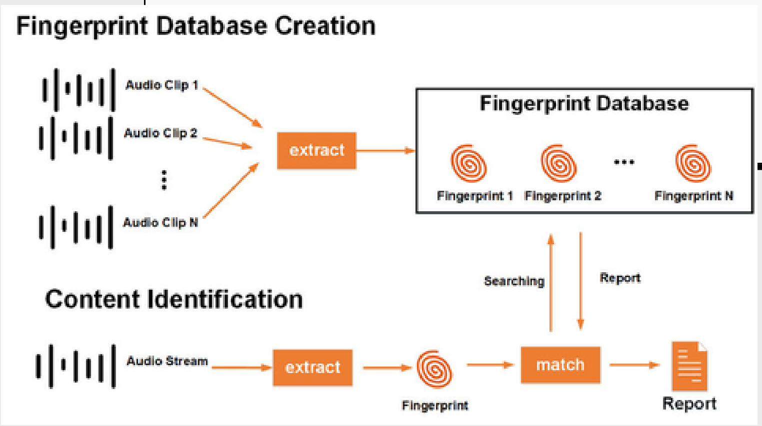


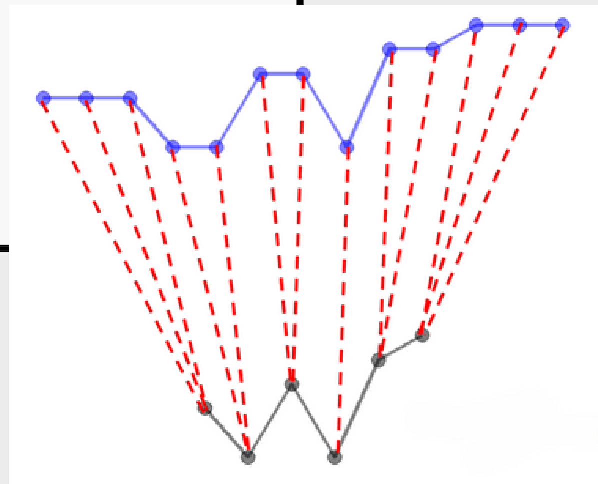
- Referencing existing melodies
- Avoiding unintentional plagiarism.

Literature Review

Mid- Semester

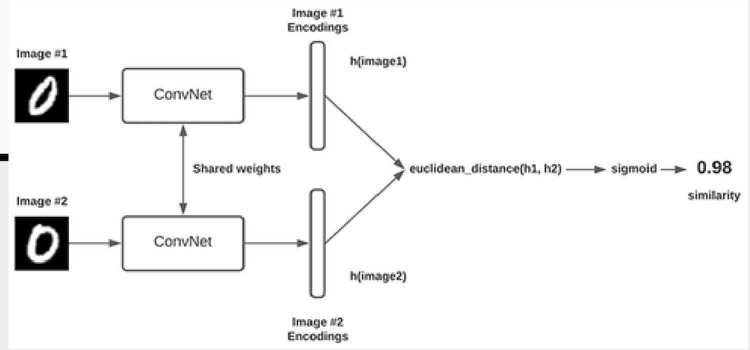
- CNN
- DTW
- Audio Fingerprinting





End- Semester

- Siamese Networks
- Feature Embeddings
- CNN



Humming-Based Song Recognition

Marar, Shreerag & Sheikh, Faisal & Swain, Drdebabrata & Joglekar, Pushkar. (2020). Humming-Based Song Recognition. 10.1007/978-981-15-1884-3_28.

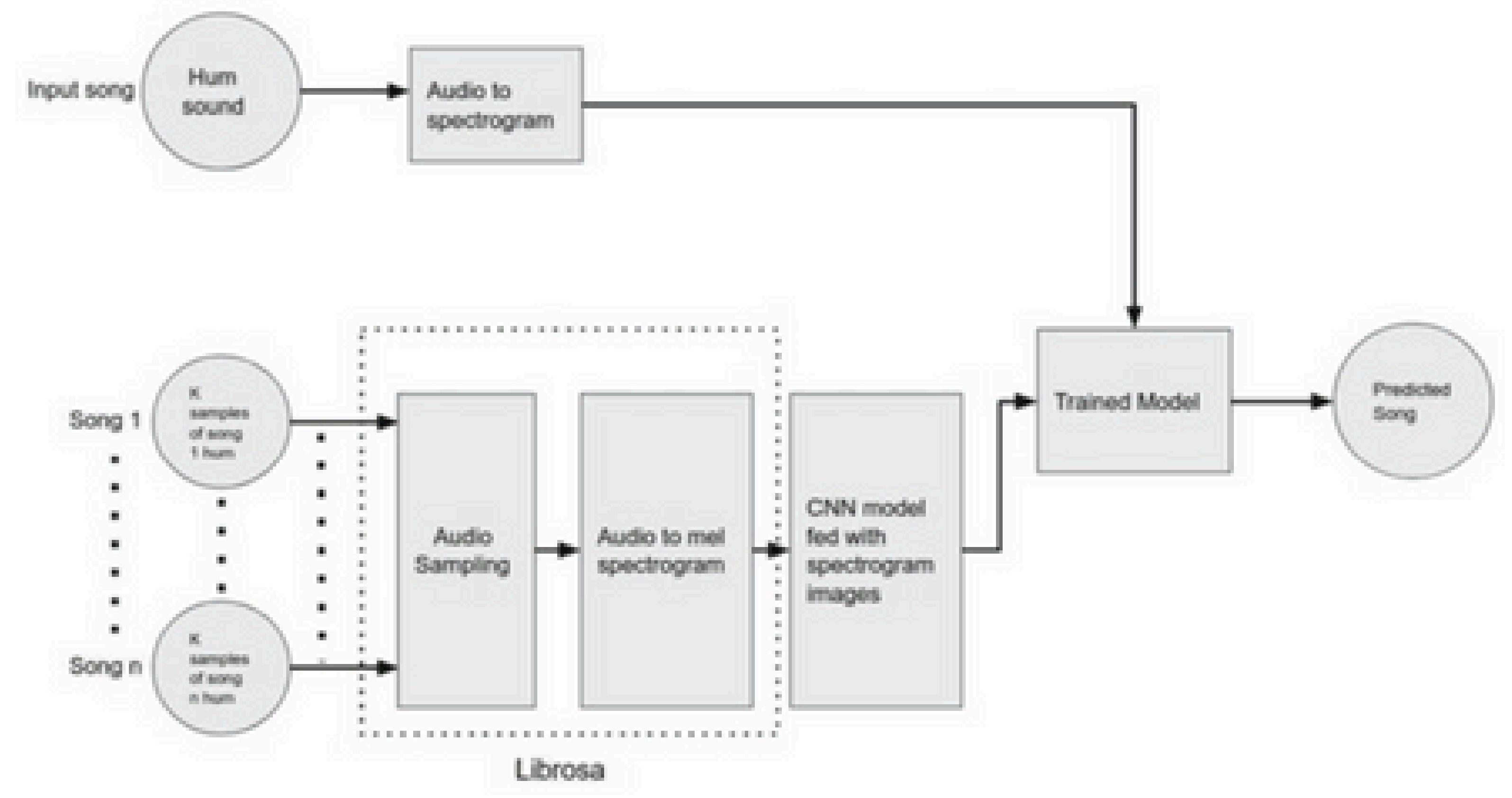


Fig. 1 Overview of the HBSR system

Training Set: 186 spectrograms

Testing Set: 48 spectrograms

6 Songs

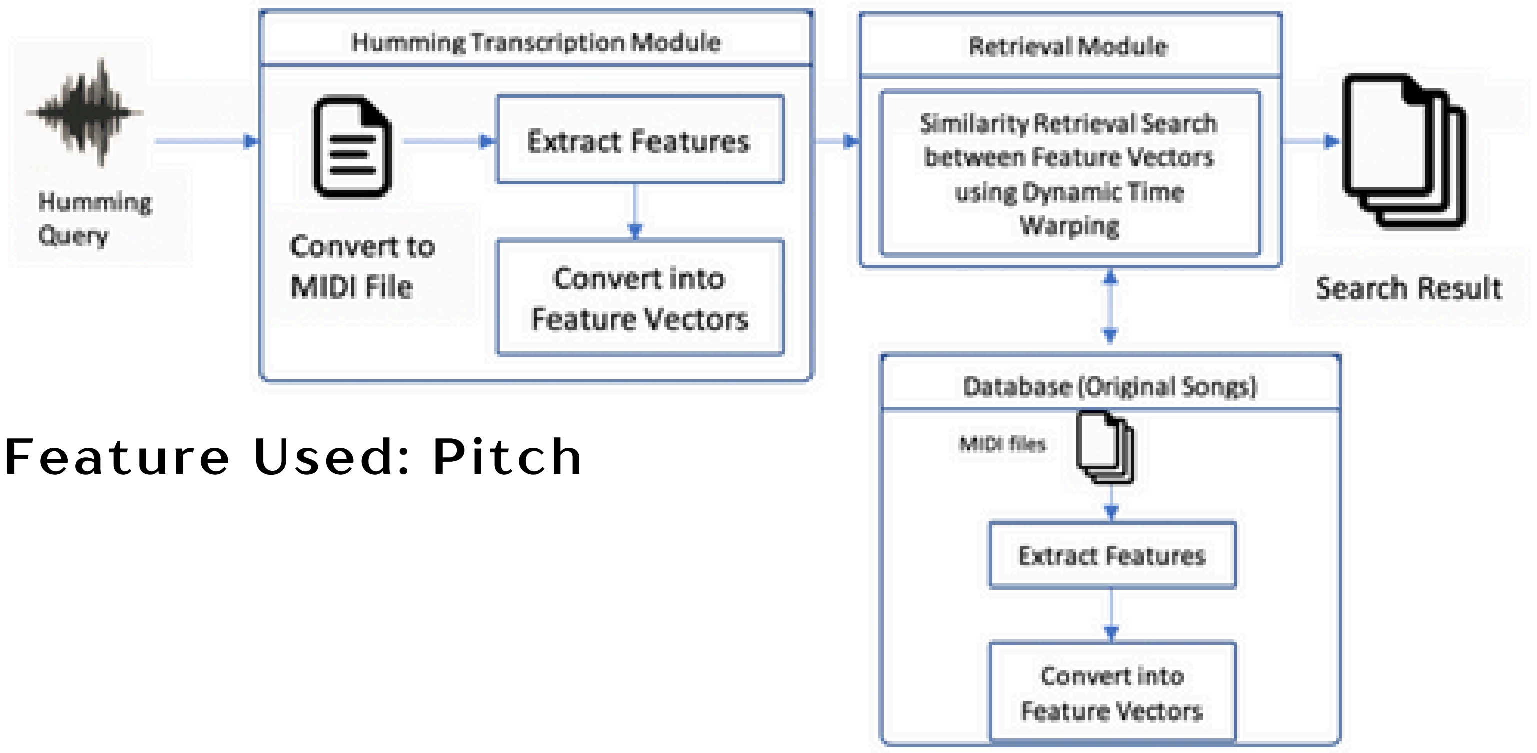
97 % accuracy

Limitations:

- Predicts **only the six songs** correctly on which the model was trained
- Multi-Class clasification hence, **not scalable**.

Music Retrieval System Using Query-by-Humming

Patel, Parth, "Music Retrieval System Using Query-by-Humming" (2019). Master's Projects. 895.
DOI: https://doi.org/10.31979/etd.mh97-77wx



Feature Used: Pitch

Figure 5: System architecture of the proposed solution.

Gender	Mean Reciprocal Rank (MRR)	
	5 songs	100 songs
Male	0.663	0.84
Female	0.66	0.80

Table 5: Average MRR of gender-based queries against different song samples.

The system is evaluated using the index known as Mean Reciprocal Rank (MRR):

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{r_i}$$

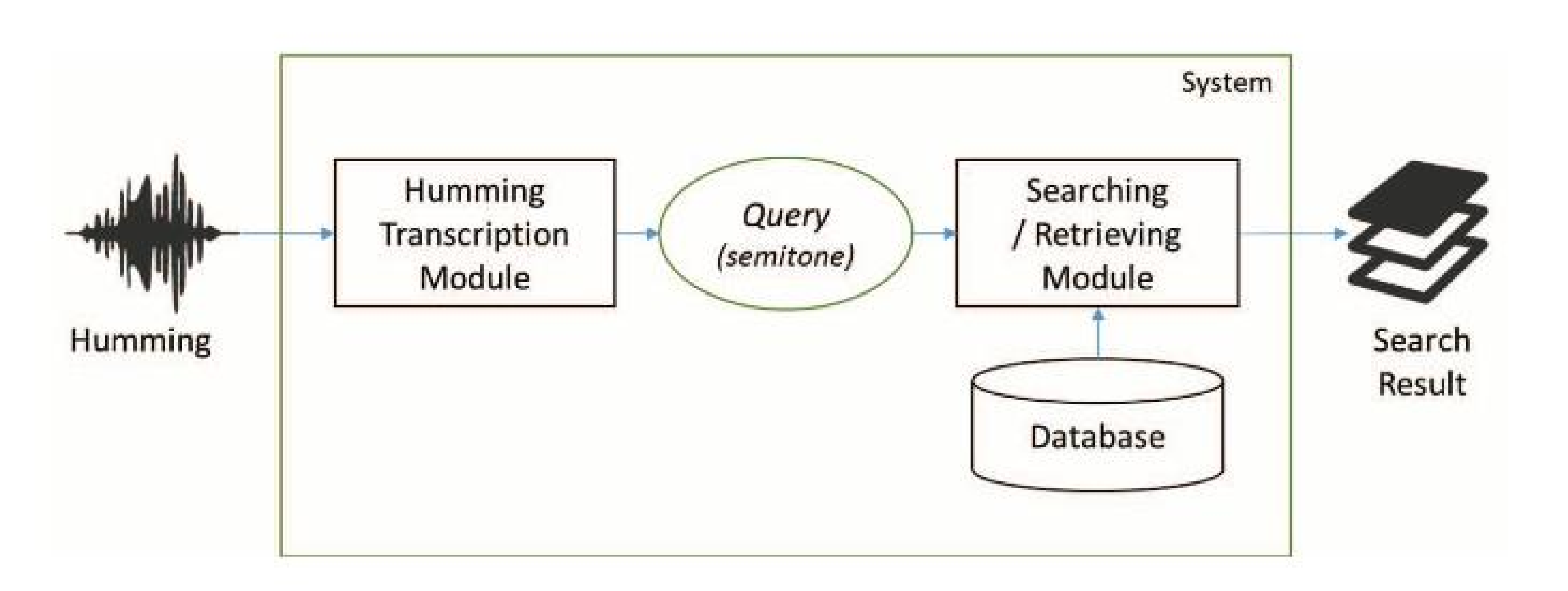
where N is the number of queries and r_i refers to the rank of the correct answer in the retrieved songs for the i-th query.

Limitations:

- Retrieval time scaled exponentially by **O(nlog(n))** according to the number of songs in the database
- No information about training data (variety of hums)

Music Information Retrieval using Query-by-Humming based on the DTW

Putri, Rifki & Lestari, Dessipuji. (2015). Music information retrieval using Query-by-humming based on the dynamic time warping. 65-70. 10.1109/ICEEI.2015.7352471.



Feature Used: Semitone extracted from pitch

Limitations:

- MRR reduced significantly when tested for songs with different keys
- Dataset contained hums from only 5 people, 50 hums each

Same key songs

MRR				
50 songs	100 songs	150 songs	200 songs	250 songs
1.00	0.98	0.98	0.97	0.97

Different key songs

MRR				
50 songs	100 songs	150 songs	200 songs	250 songs
0.23	0.18	0.17	0.17	0.17

Towards Cover Song Detection with Siamese Convolutional Neural Networks

Stamenovic, Marko. (2020). Towards Cover Song Detection with Siamese Convolutional Neural Networks.

Figure 1. Network architecture and hyperparameters of the proposed algorithm.

Precision of 65.0%

Feature Used: Q-Power Spectrogram

- Dataset contains 24,986 cover-song pairs
- Objective of the model- Cover Song Retrieval (not using hums)

Dataset

MLEnd Hums and Whistles dataset

Developed by students at the School of Electronic Engineering and Computer Science, Queen Mary University of London

Audio Files

6613

8 songs

- Humming audio files with accurate labels
- 73% hums and 27% whistles
- 235 different interpreters of various nationalities
- Meta Data: Interpreter, Song Label, Interpretation (Hum/Whistle)
- Recorded using simple microphone under normal conditions

Lot of data for each song, but no diversification!

Data Collection & Augmentation

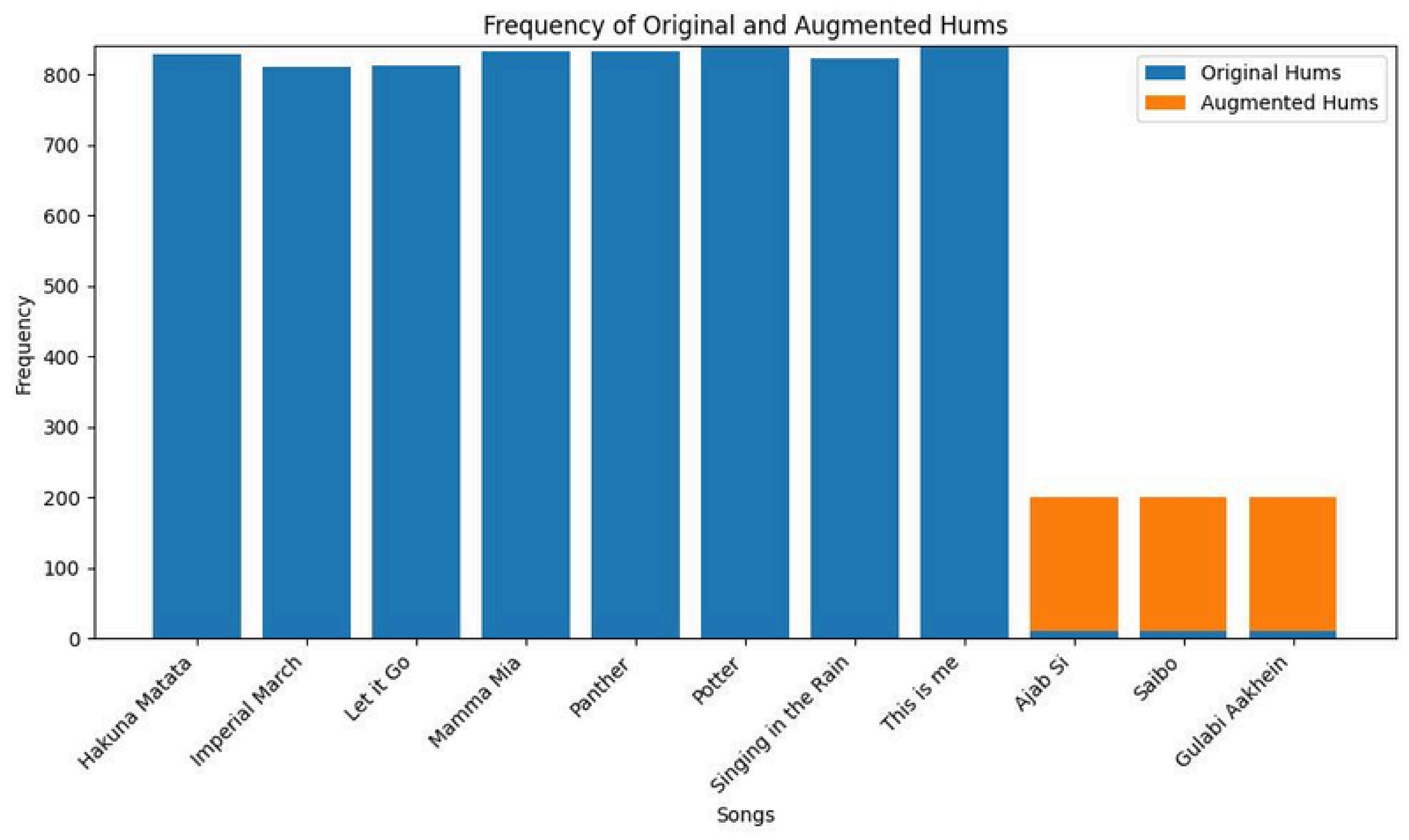
4 Songs, 40 Hums

Songs chosen for collection:

- From cluster with less hums
- Easy to hum
- Well Known
- Present in the Dataset
- 3 Hindi, 1 English
- From different decades

5 Augmented Hums per Hum

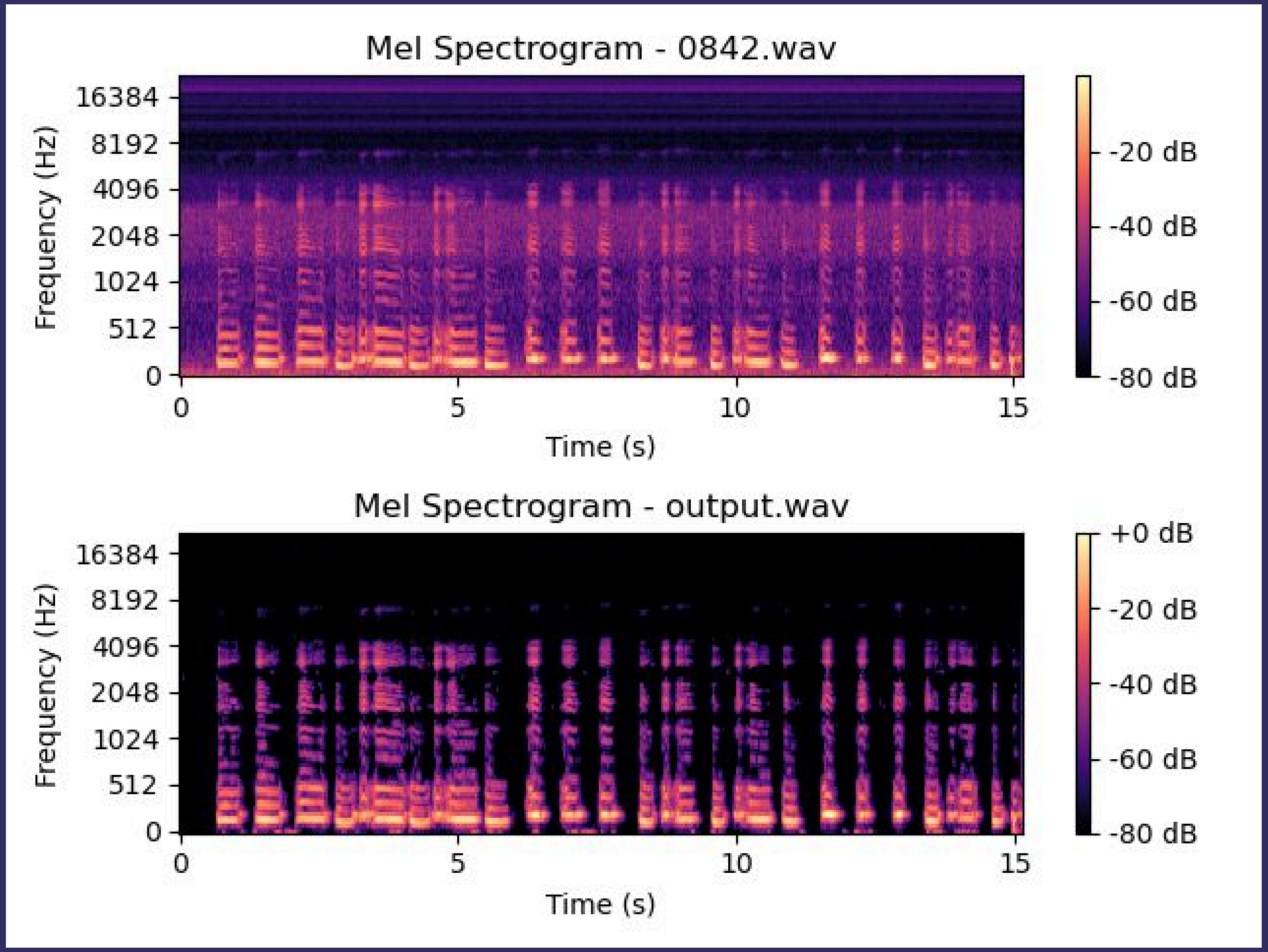
- Noise Addition: Random noise with a variable standard deviation (0.005 to 0.02) using TensorFlow
- Pitch Shifting: Shifted pitch by −2 to 2 semitones using librosa
- Time Stretching & Speed Variation: Altered playback speed by 0.8 to 1.2 times with pydub

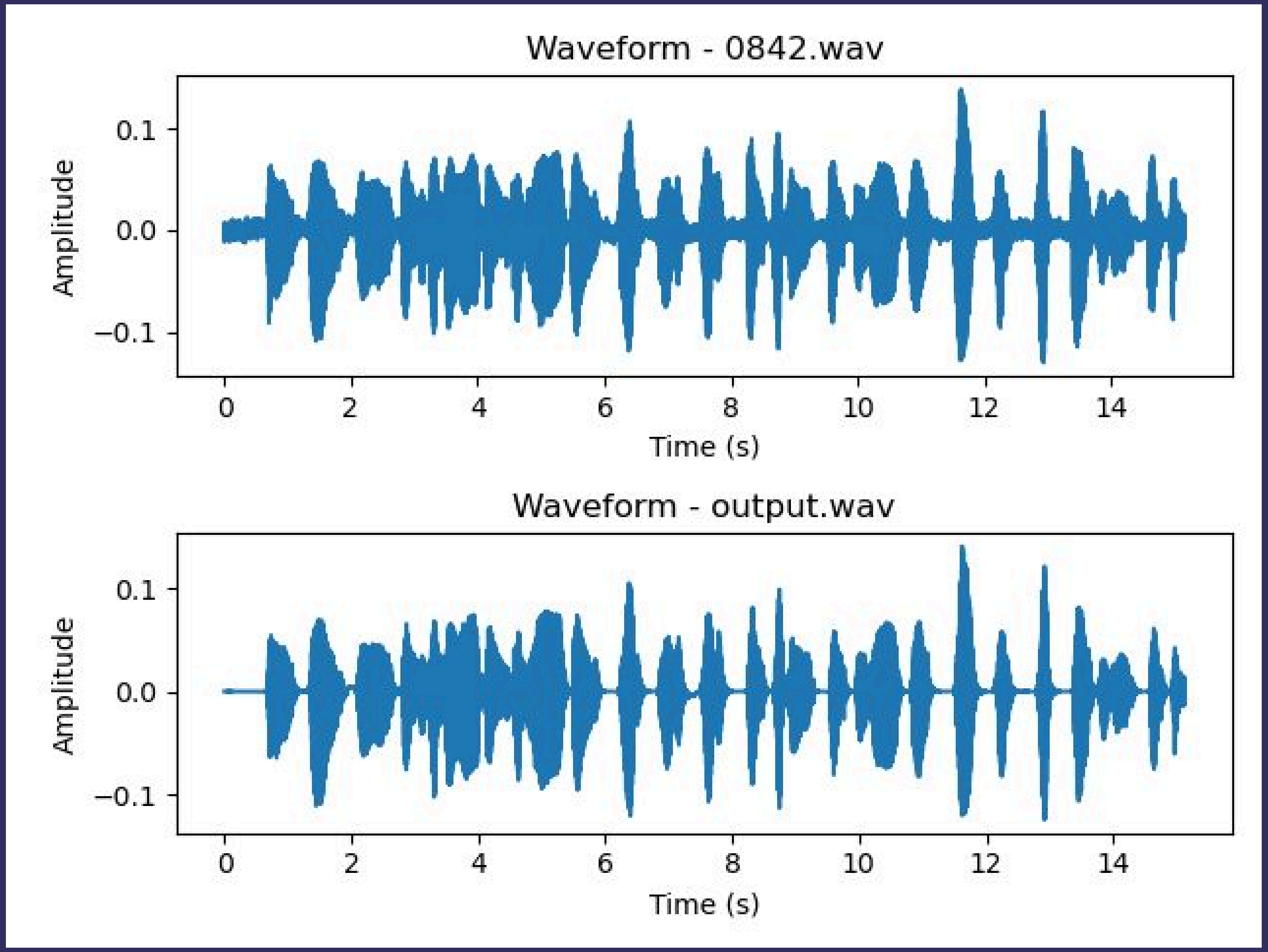


Frequency of Original and Augmented Hums
Original Hums
Augmented Hums
Frequency
0
100
200
300
400
500
600
700
800
Hakuna Matata
Imperial March
Let it Go
Mamma Mia
Panther
Potter
Singing in the Rain
This is me
Ajab Si
Saibo
Gulabi Aakhein
Songs

Data Preprocessing

1. **Noise Reduction: Using 'sox' , to remove low intensity background noise on hums**





Data Preprocessing

2. Trimming hum length to 15 seconds

- If silences are present, remove leading & trailing silences
- Else, remove trailing part
- For shorter songs, add padding at the end

3. Sampling Rate set to 16 kHz

4. Database of 50 songs, 15 second portions (chorus of song)

Features Extraction

Hums

For Similarity:

- **MFCC**– 20 coefficients
- Number of time frames: **500**

Songs

For Clustering:

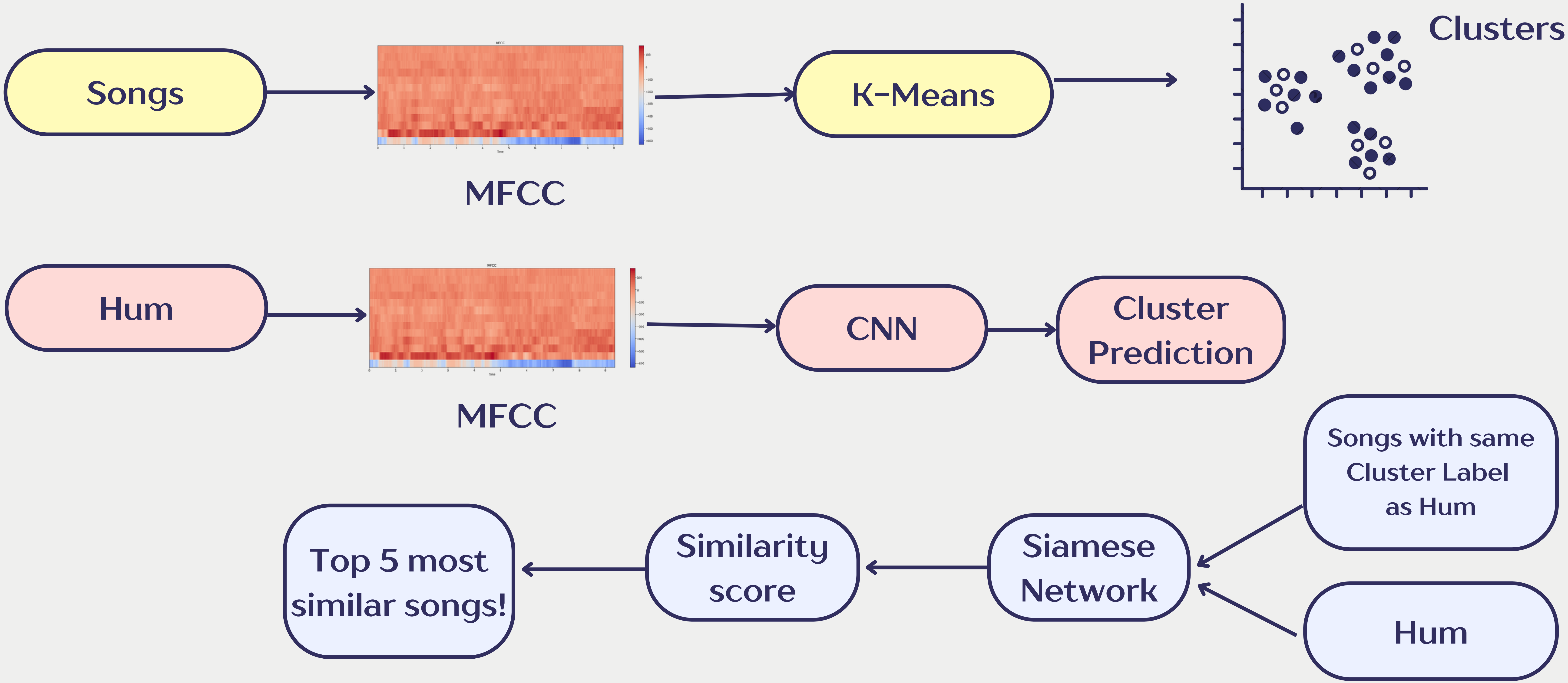
- Combination of **MFCC, Mel, Spectral, Pitch & Chroma** features
- **Mean** of all taken across timeframe
- Dimensionality Reduction through PCA

For Similarity:

- **MFCC**– 20 coefficients
- Number of time frames: **500**

ML Methodology

Model Architecture

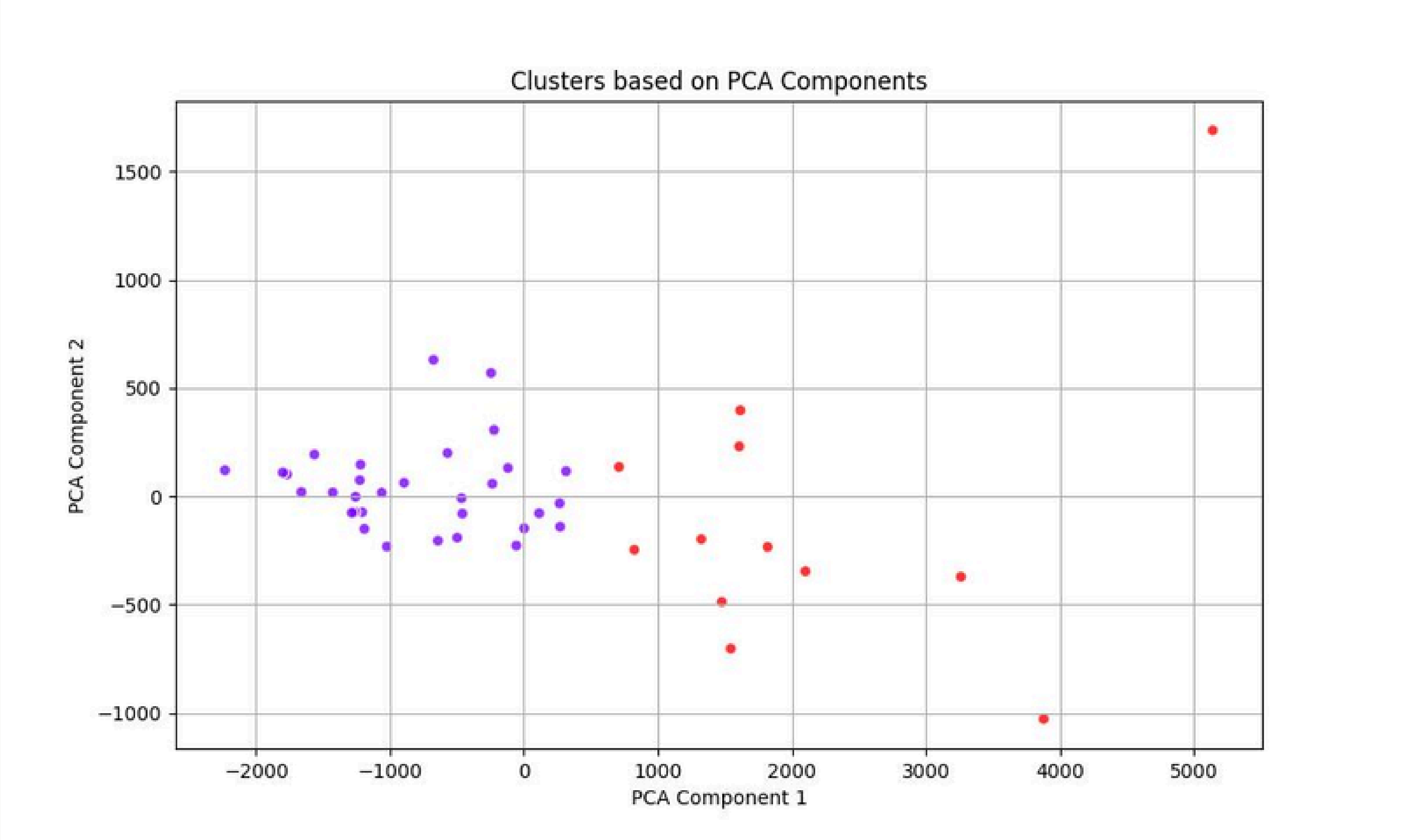


1. Clustering of Songs using K– Means

(Silhouette scores)

No. of Clusters	All Features(198)	Mfcc & Spectral Features (33)	Mfcc, Spectral, Chroma etc. (78)	Only Mfcc (mean, variance std.– 60)
2	0.50	0.53	0.53	0.62
3	0.52	0.54	0.54	0.55
4	0.45	0.52	0.52	0.58
5	0.48	0.57	0.57	0.53

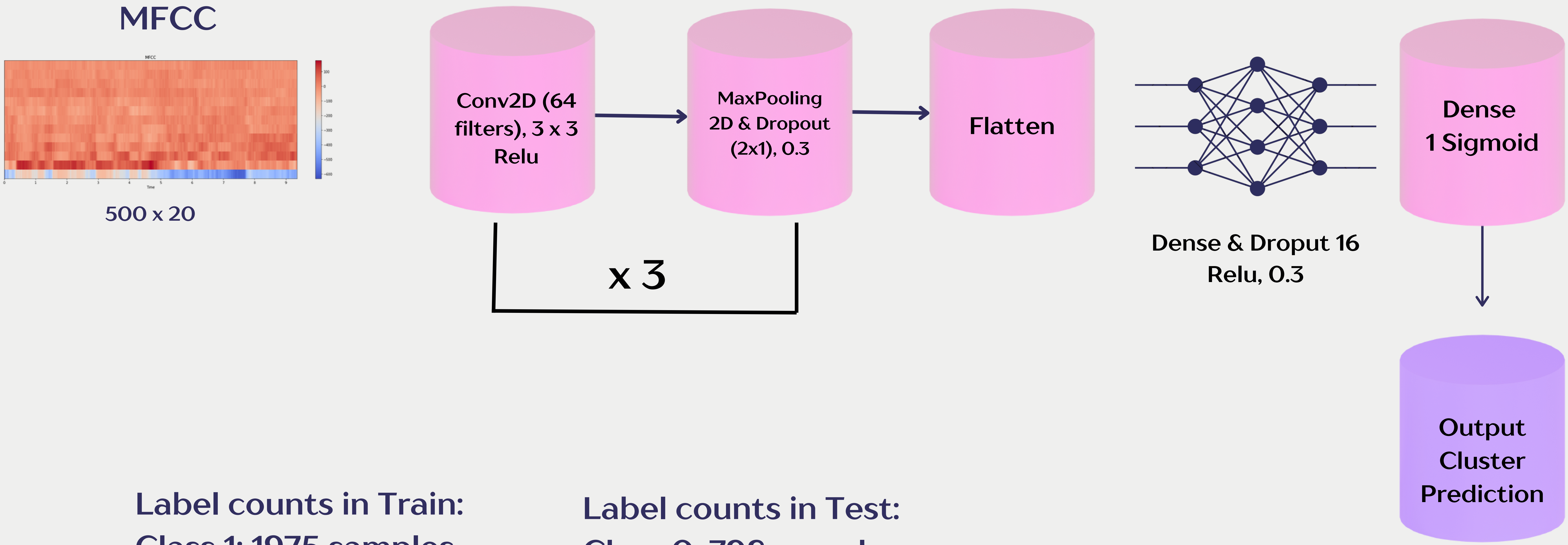
1. Clustering of Songs using K– Means



Class 0: 34
Class 1: 16

Total number of songs in the database: 50

2. Classification of Hums using CNN



Label counts in Train:
Class 1: 1975 samples
Class 0: 3313 samples

Label counts in Test:
Class 0: 798 samples
Class 1: 525 samples

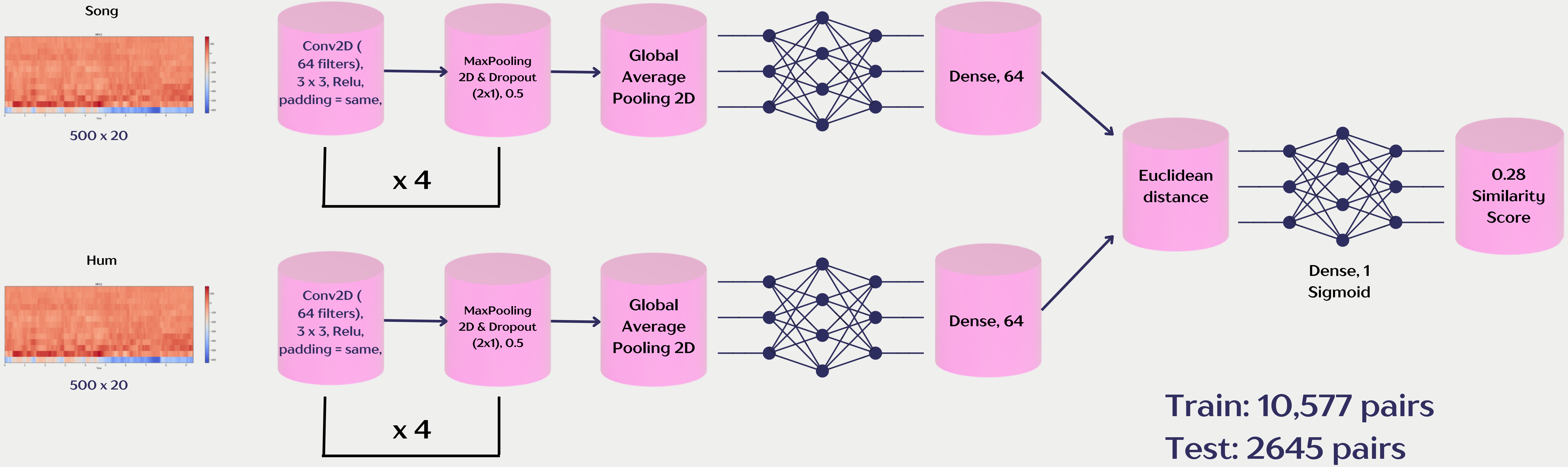
2. Classification of Hums using CNN

Confusion Matrix

	Predicted 0	Predicted 1
Actual 0	712	86
Actual 1	62	463

Balanced Accuracy: 0.887
F1- score 0.9

3. Siamese Network



Train: 10,577 pairs
Test: 2645 pairs

3 pairs per hum: 1 similar pair, 2 dissimilar pairs

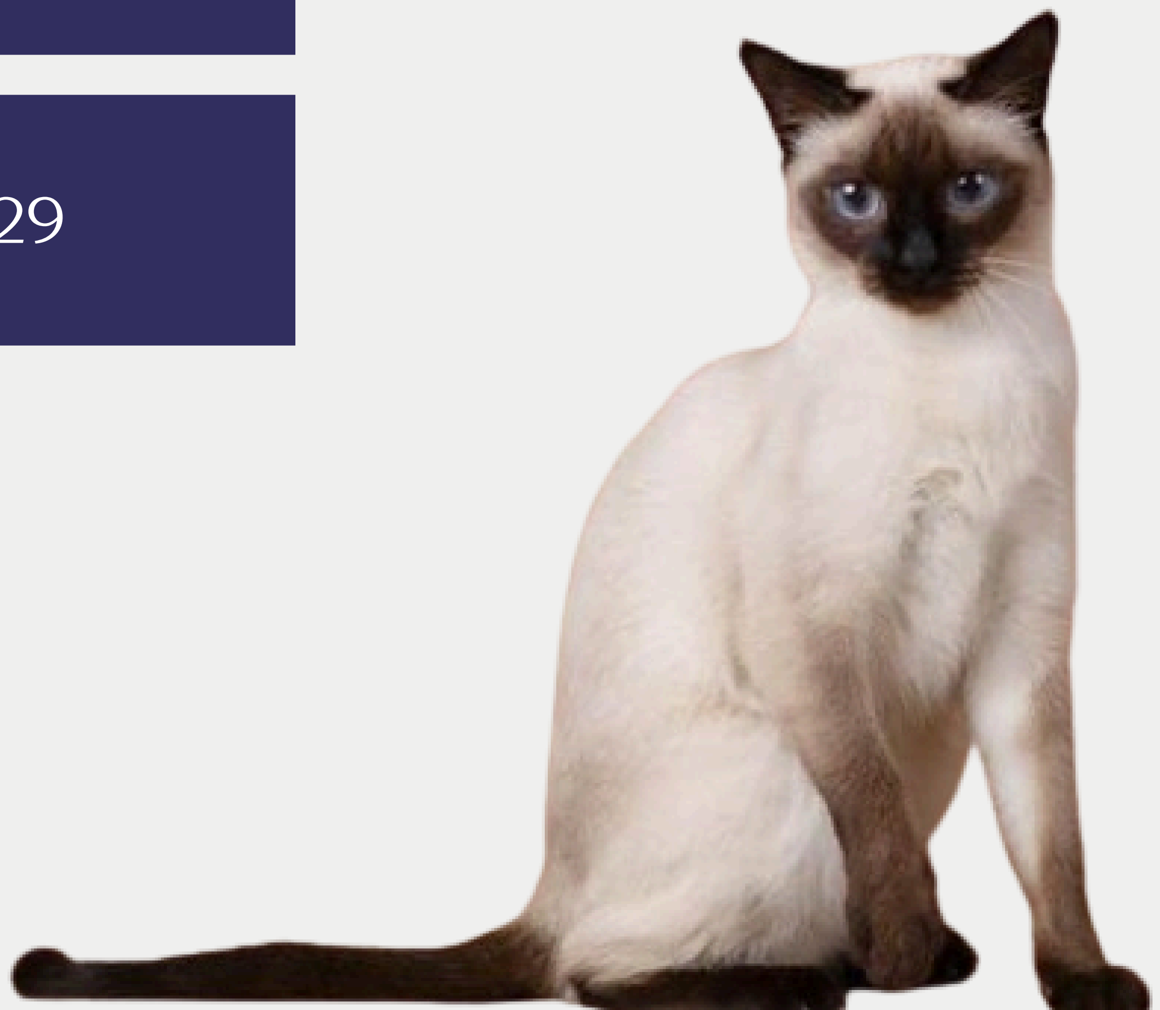
3. Siamese Network

Confusion Matrix

	Predicted 0	Predicted 1
Actual 0	1614	703
Actual 1	26	1129

Test Accuracy: 0.79

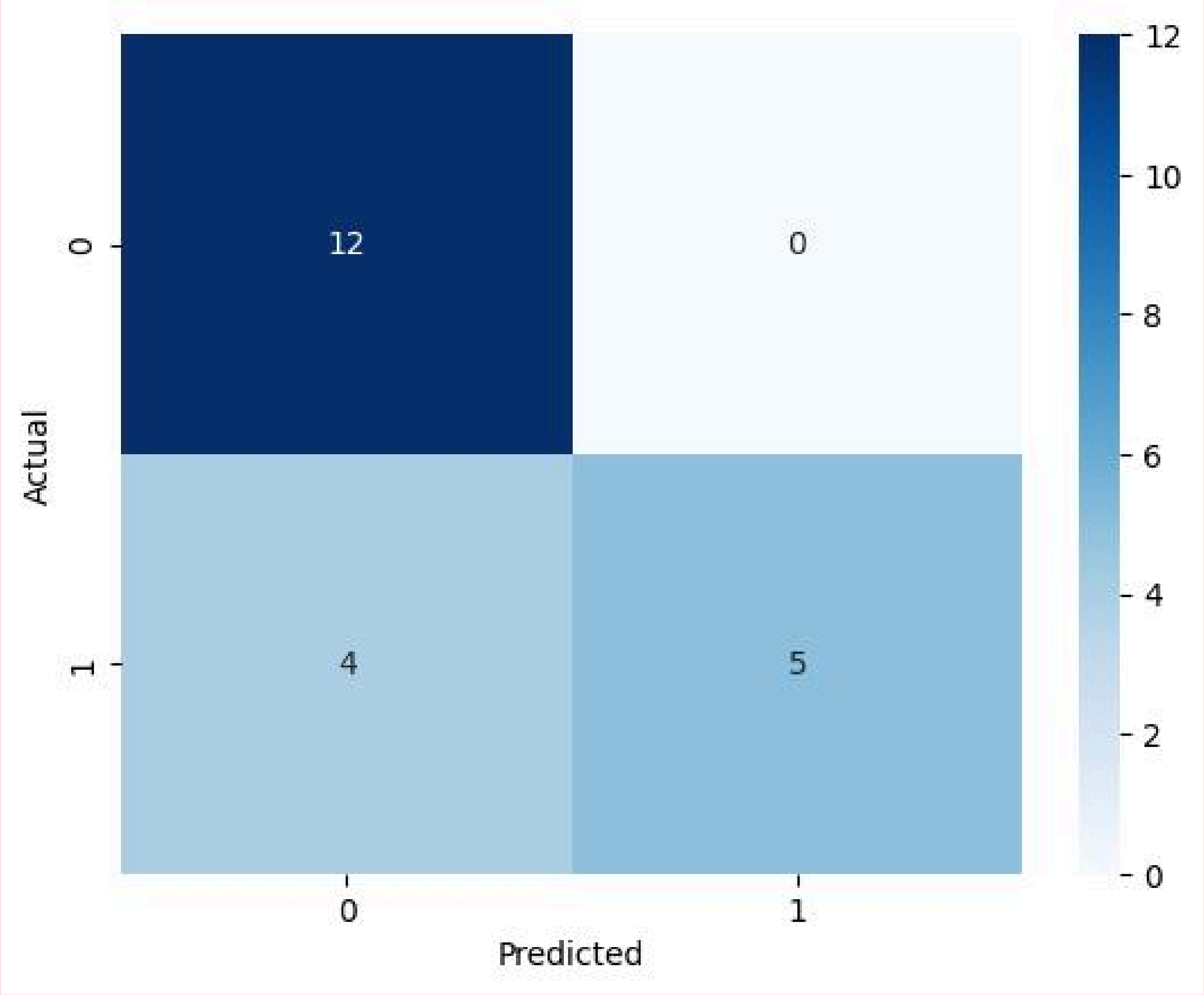
Balanced Accuracy: 0.836



Performance Metrics

21 randomly selected hums

1. Cluster classification



2. Retrieval

<u>Search Space: 25 songs</u>

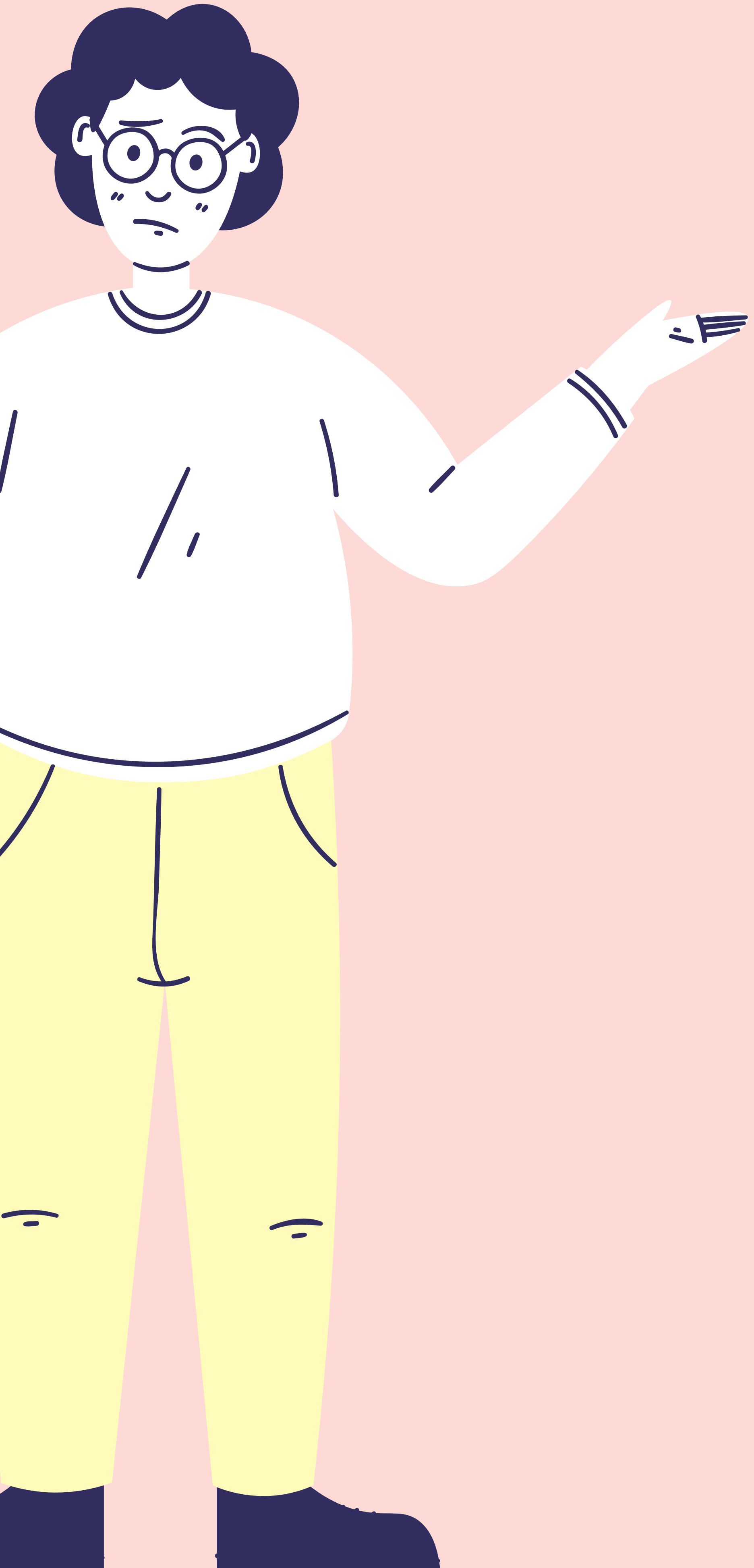
Top 1 Correct Prediction: 0.33, 7 out of 21
Top 3 Correct Prediction: 0.476, 10 out of 21
Top 5 Correct Prediction: 0.741, 15 out of 21

MRR: 0.62

<u>Search Space: 10 songs</u>

Top 1 Correct Prediction: 0.619, 13 out of 21
Top 3 Correct Prediction: 0.761, 16 out of 21
Top 5 Correct Prediction: 0.809, 17 out of 21

MRR: 0.88



Challenges Faced

01 **Less Data:** Even though we tried to collect and augment hums for new songs, we couldn't achieve enough to be able to train the model to generalize it to any song in the dataset

02 **No backbone:** Even with existing work done on this topic, we couldn't find papers giving a detailed walkthrough of their process. The model we created was entirely new, and unexplored.

03 Not just any part of the song can be used for prediction, it is only the 15 second clips that have been used in the dataset that can be used (Based on the assumption that people only hum the chorus)

04 **Hardware Constraints:** We had to reduce our sampling rate and we could not use our model on Mel spectrograms.

Deployability & Hurdles

- Model only works on songs in the training set, hence, it isn’t deployable. It will need more training and adjusments to deal with newer data.

- Currently our system is working well, as we do not have too many songs. However, as the size of the music library grows, the search space and computational requirements will go up. So we would need to improve the clustering algorithm.

End Goal:

1) A Mobile Application
2) API endpoint for integration with other music services

References

1] *https://link.springer.com/chapter/10.1007/978-981-15-1884-3_28*

2] *https://blog.research.google/2020/11/the-machine-learning-behind-hum-to.html?m=1*

3] *https://www.cs.cornell.edu/zeno/papers/humming/humming.pdf*

4] *https://dl.ucsc.cmb.ac.lk/jspui/bitstream/123456789/4616/1/2018%20MCS%20001.pdf*

5] *https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1895&context=etd_projects*

6] *https://www.kaggle.com/datasets/jesusrequena/mlend-hums-and-whistles?select=MLEndHWD_Audio_Attributes.csv*

7] *https://www.toptal.com/algorithms/shazam-it-music-processing-fingerprinting-and-recognition*

8] *Marar, Shreerag & Sheikh, Faisal & Swain, Drdebabrata & Joglekar, Pushkar. (2020). Humming-Based Song Recognition. 10.1007/978-981-15-1884-3_28.*

9] *Patel, Parth, "Music Retrieval System Using Query-by-Humming" (2019). Master's Projects. 895. DOI: https://doi.org/10.31979/etd.mh97-77wx*

10] *Putri, Rifki & Lestari, Dessipuji. (2015). Music information retrieval using Query-by-humming based on the dynamic time warping. 65-70. 10.1109/ICEEI.2015.7352471*

11] *Stamenovic, Marko. (2020). Towards Cover Song Detection with Siamese Convolutional Neural Networks.*

Thank You!