SPEECH PROCESSING REPORT

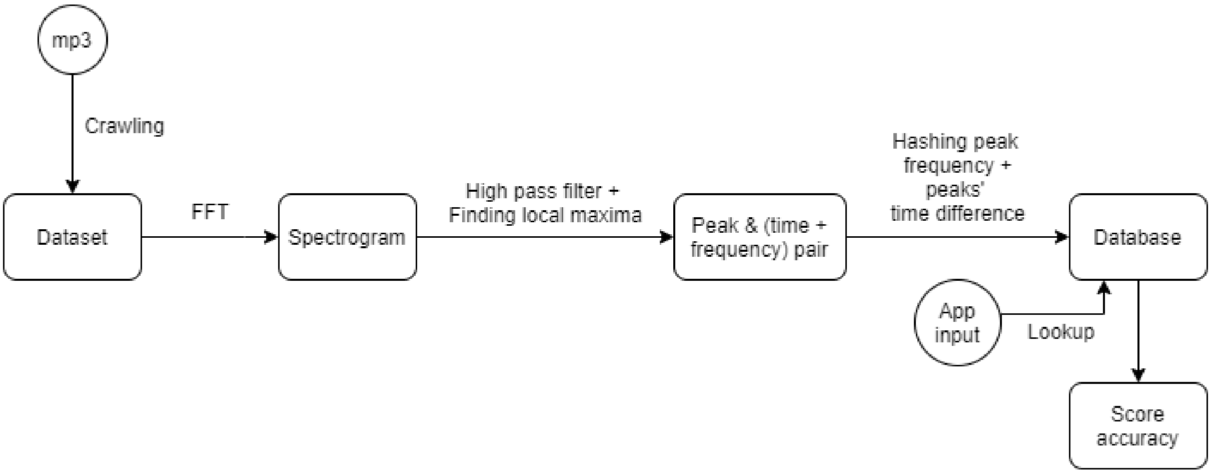
SONG RECOGNITION & VOICE STYLE FILTER

Đặng Nhật Linh
17021283

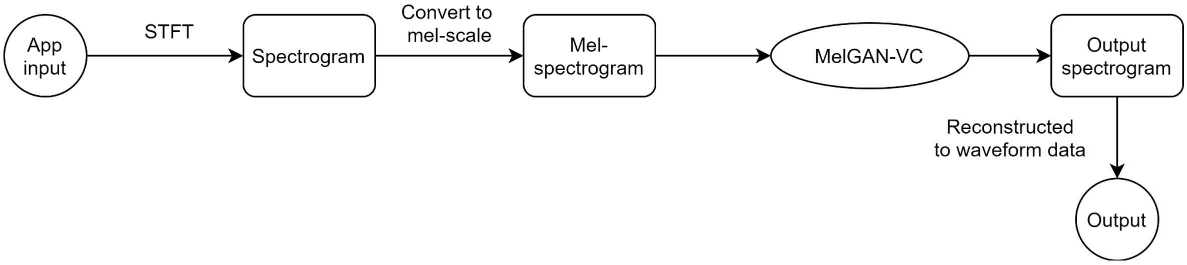
Phạm Thái Sơn
17021330

OVERVIEW

Song recognition


mp3
Crawling
Dataset
FFT
Spectrogram
High pass filter + Finding local maxima
Peak & (time + frequency) pair
Hashing peak frequency + peaks' time difference
Database
App input
Lookup
Score accuracy

Voice style filter



Features & Tech Stack

Algorithm: Python, numpy & scipy for spectrogram processing, tensorflow & pytorch for MelGAN-VC

Demo application: Django & React

SONG RECOGNITION

Approach 1:

Using a frame-base approach to capture fingerprints for the dataset + a sliding window to match the input to the dataset.

A standard procedure for this approach would be to separate waveforms into multiple frames, make them the feature vector and then match the input data with a database of those vectors.

Cons:

- Does not account for channel variations.
- Sensitive to offset similarity across a dataset with multiple songs containing feature vector resemblances.
- Inefficient computation time, especially with bigger datasets.
- Very weak result in cases with added noise.

Conclusion:

The simplest algorithm to implement in feature choosing to fingerprint, a frame-base approach is only effective in a very strict input environment. Sliding windows is an intuitive matching way to quickly build and verify the result of the previous steps, but more tweaking is needed for real-time application.

Approach 2:

Using Philips algorithm [1] to filter prominent features from the spectrogram for capturing fingerprints for the input dataset.

To sum up, the Philips algorithm compresses the spectrogram as much as possible, and then looks out for changes in time and frequency. The fingerprint would store the changes only when they are above a set threshold amount.

Cons:

- Not very resistant to continuous noise, rendering one of the most frequent use-case of the application (finding music in public settings) inappropriately not reachable.

Conclusion:

Although proving to be exceedingly immune to strong, high-pitch noise burst and pitch shift, the algorithm fails to deliver good results on extended turbulence cases and thus was not chosen.

Approach 3:

Using Shazam's approach to solve the previously mentioned problems.

This approach utilizes the following design:

- Peak Finding (in Spectrogram): Instead of comparing changes within the given spectrogram, Shazam's uses an architecture of peak (the greatest time - frequency pair in a neighborhood) time difference connection, with the end result visually referred to as similar to a "spider web".
- Hashing fingerprint: The above "spider web" is then used to create an unique hash value for each fingerprint. This ensures that all the remaining information in the fingerprint is stored using minimum memory.

Result:

For testing purposes, 100 songs from Billboard's Hot-100 Decade-end list were crawled and loaded into the database.

Audio segments extracted from original files in the database achieved a 100% accuracy record.

For noise testing, a script was created to separate 100 different song records over the microphone, with random noise in the background, into random n-second sequences (to a total of 350 test sequences). Here is our accuracy result:

n value (second(s))	Accuracy
1	58.75%
2	94.50%
3	98.25%
4	98.50%
5	100.0%

Conclusion:

The result proves to be remarkably efficient for our current use case (low-to-moderate background noise, no extreme peak, small dataset for testing purpose).

For a more robust solution, other machine learning methods like Intrasonics [3] shall be looked into to ensure more generalized feature extractions. At the same time, different hash algorithms and database table configurations might be analyzed to achieve better speed/storage trade-off, and to encounter one of the current major disadvantages - not being able to process a song with shifted playback speed.

VOICE STYLE FILTER

Approach:

Using MelGAN-VC [4] to convert male voice to female.

Using STFT (Short-time Fourier Transform), the input audio file is converted to a spectrogram. The subsequent product is transformed into a decibel scale, before remodeled into a mel spectrogram.

MelGAN-VC is a generative adversarial network and thus has a split-feed-concatenate training pipeline - the resulting samples are fed to a discriminator to translate arbitrary length inputs without error.

Further explanation of the model architecture is described in the Technical Details section.

Training:

The model was trained on the dataset provided by INT3411_20 class of 2021. The dataset is divided into male and female sections for both domains. The model is trained for 300 epochs with a learning rate of 0.0002.

Conclusion:

Although the results were unsatisfactory, it still serves as a nice experiment in the style transfer area field of Deep learning, particularly in how image-to-image translation can be used to do audio processing as well.

TECHNICAL DETAILS

Refer to this slide for details.

Reference:

[1] J. Haitsma, T. Kalker (2002). A Highly Robust Audio Fingerprinting System. Proc. International Conf. Music Information Retrieval.

[2] Piotr Indyk, Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality.

[3] A. Wang (2006), The Shazam music recognition service, Comm. ACM 49(8), 44-48.

[4] M. Pasini (2019), MelGAN-VC: Voice Conversion and Audio Style Transfer on arbitrarily long samples using Spectrograms, arXiv preprint arXiv:1910.03713.