Audio Fingerprinting

Anup Singh

GHENT UNIVERSITY

WiSSAP 2023

Tutorial Outline

1. Introduction
2. Fingerprinting
3. Indexing
4. Literature Review

1

Introduction

What’s this song?

Audio fingerprinting

- Generates content-based summary of an audio signal.
- Allows efficient storage and retrieval of similar audio in large database.

Applications

Music
Identification

Broadcast
monitoring

Second screen
applications

Challenges

Robustness

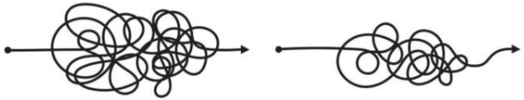
against audio distortions such as noise and reverberation

Search speed

to expedite the retrieval process to enhance user experience

Computational viable

for practical deployment



Modules

1

Audio Processing

Includes audio segmentation and pre-processing

2

Representation Learning

Generates compact and robust audio representations

3

Indexing

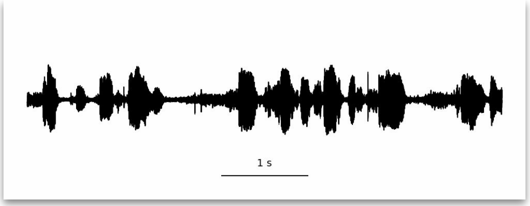
Enables fast retrieval

2

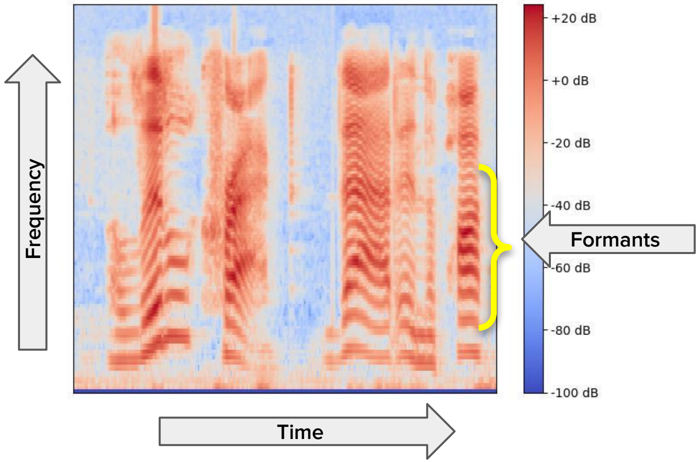
Fingerprinting

01 Audio Processing

- **Waveform:** A digitized audio signal.
- **Preprocessing:** Resampling, filtering, audio normalization
- **Time-frequency representation:** 2-D matrix representation of frequency contents of an audio signal over time.
    - Spectrograms and its variants
    - Very few audio fingerprinting approaches directly process waveforms



source: https://jvbalen.github.io/notes/waveform.html



02 Representation Learning

Transform audio segments into low-dimensional vectors.

Rule-based

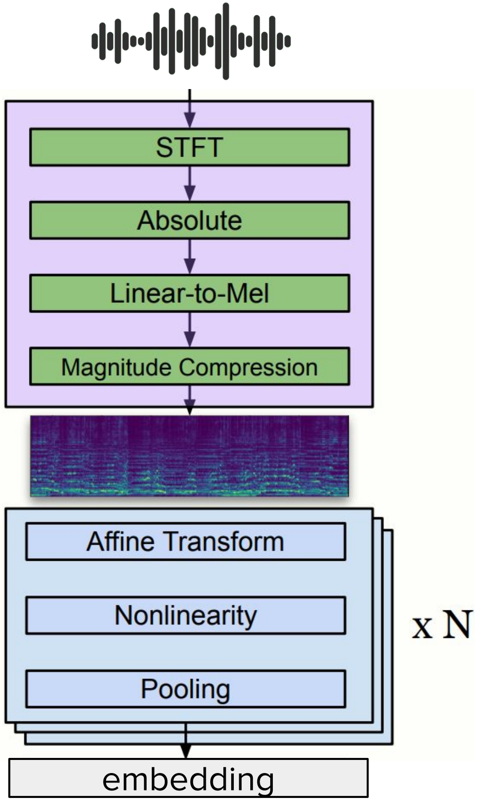
- Handcrafted features
- Not resilient against high distortion environments.
- Exemplar:
  - Shazam: peak based
  - Philips: energy-difference-based
  - Waveprint: top-k wavelets

Learning-based

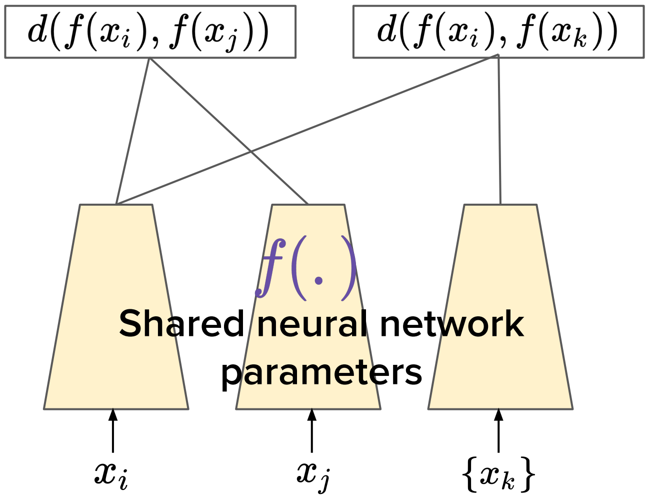
- Real-valued features
- Highly resilient
- Exemplar:
  - RNN-based
  - CNN-based
  - Transformer-based

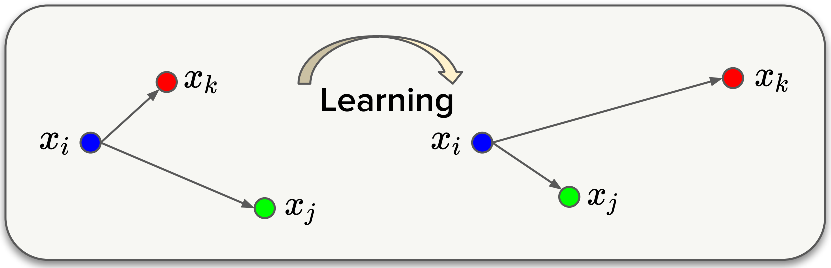
02 Deep embeddings

- Deep neural networks $f(.)$ as nonlinear embedding function.
- Gradient descent optimization
- Deep embeddings⟷fingerprints



02 Self-supervised learning



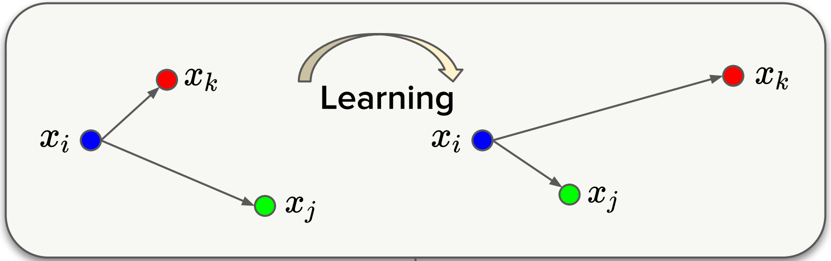


x_i and x_j are relevant, not x_k

- x_i : anchor sample
- x_j : positive sample
- x_k : negative sample

02 Self-supervised learning

- **Input:** log Mel spectrogram (segment): x_i
- **Model:** CNN
- **Training:** Contrastive learning with
    - Distorted audio: x_j
    - Any other audio: x_k



$$\mathcal{L} = -\log \frac{e^{d(f(x_i),f(x_j))/\tau}}{\sum_{i\neq j} e^{d(f(x_i),f(x_j))/\tau}}$$

Contrastive Loss

02 Contrastive Training

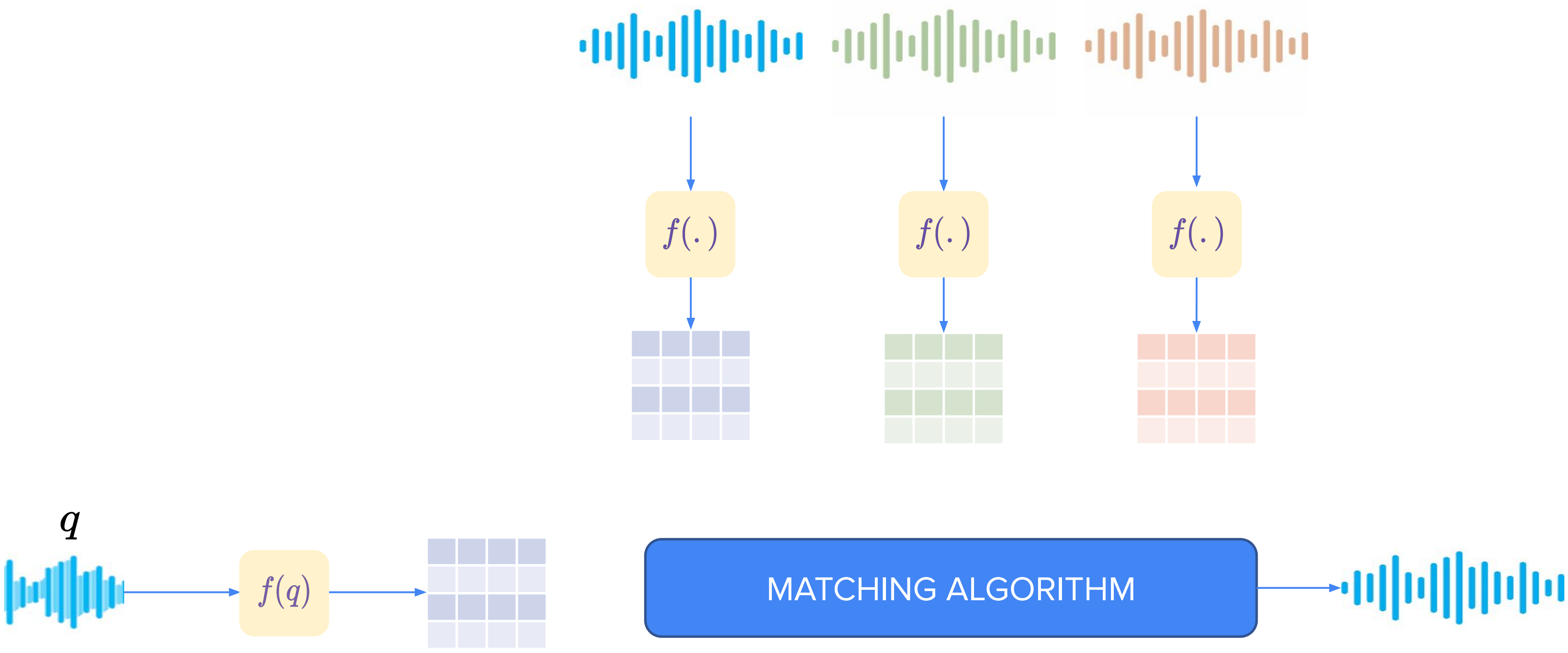
- Distortions are randomly applied:
  - **Real-world noises**: 0-25 dB SNR
  - **Reverberation:** RIRs with t_{60} levels ranging from 0.1s - 0.8s
  - **Time offset:** 50 ms

Coding time!

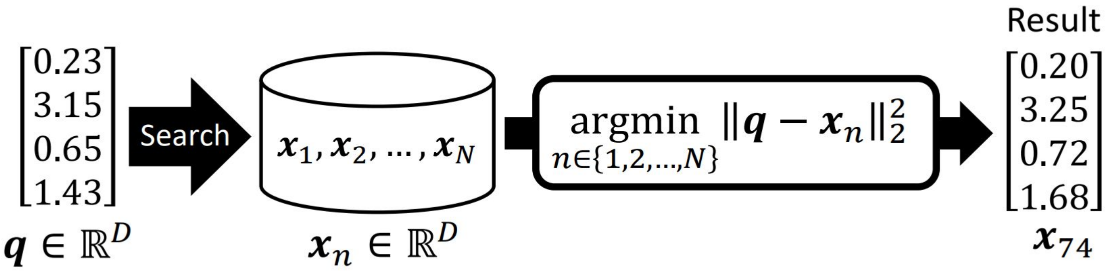
3

Indexing

03 Retrieval



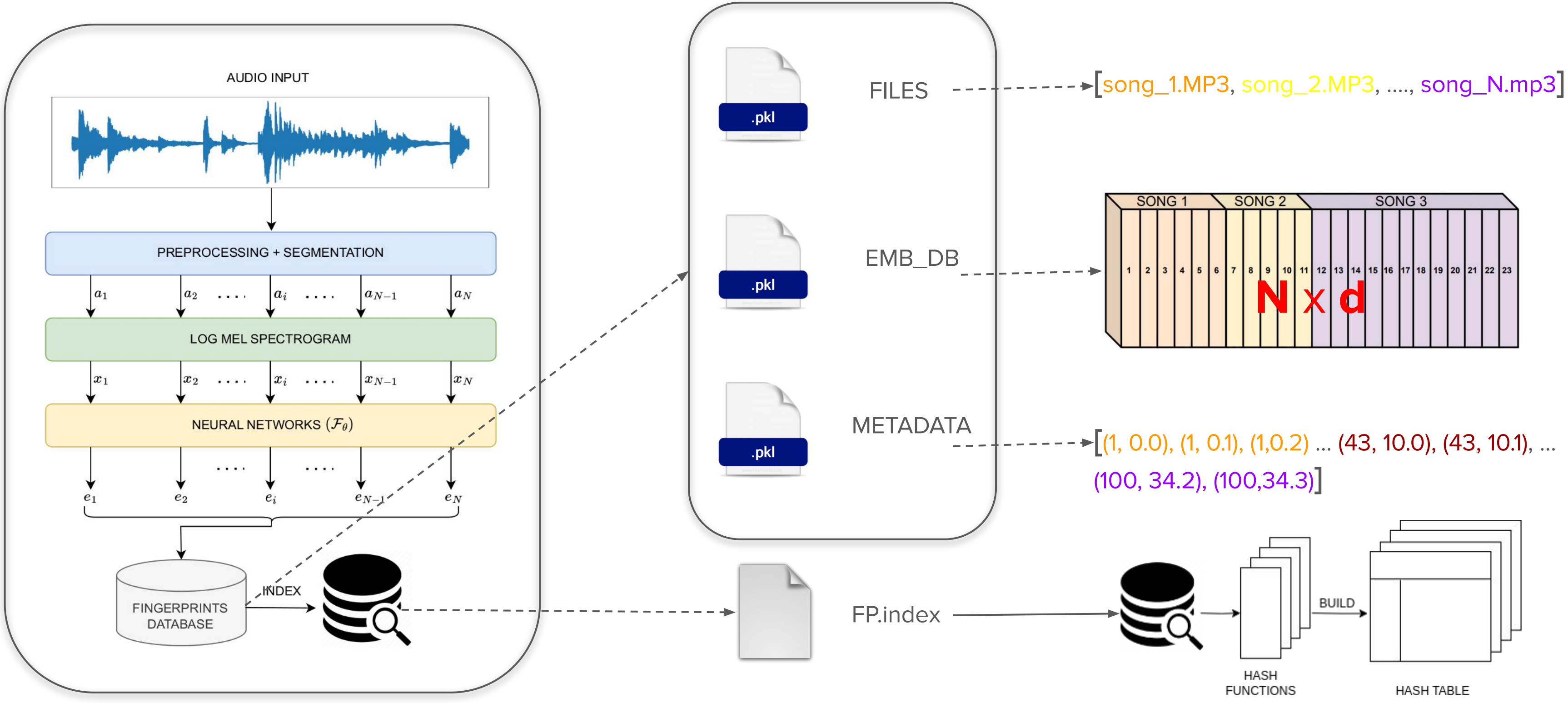
03 Retrieval



- N D-dim database vectors
- **Task:** For the given query, find the closest vector from the database
- **Linear scan:** O(ND), slow
- **ANN:** Sublinear query time
  - don’t necessarily have to be exact neighbors
  - Trade off: runtime, accuracy and memory-consumption

- ANN algorithms: LSH (hash-based), PQ (vector quantization-based), HNSW (graph based)

03 Indexing



AUDIO INPUT
PREPROCESSING + SEGMENTATION
a_1 a_2 .... a_i .... a_{N-1} a_N
LOG MEL SPECTROGRAM
x_1 x_2 .... x_i .... x_{N-1} x_N
NEURAL NETWORKS ($\mathcal{F}_\theta$)
e_1 e_2 .... e_i .... e_{N-1} e_N
FINGERPRINTS DATABASE
INDEX
.pkl
FILES
[song_1.MP3, song_2.MP3, ..., song_N.mp3]
.pkl
EMB_DB
SONG 1
SONG 2
SONG 3
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
N x d
.pkl
METADATA
[(1, 0.0), (1, 0.1), (1,0.2) ... (43, 10.0), (43, 10.1), ...
(100, 34.2), (100,34.3)]
FP.index
BUILD
HASH FUNCTIONS
HASH TABLE

03 Data

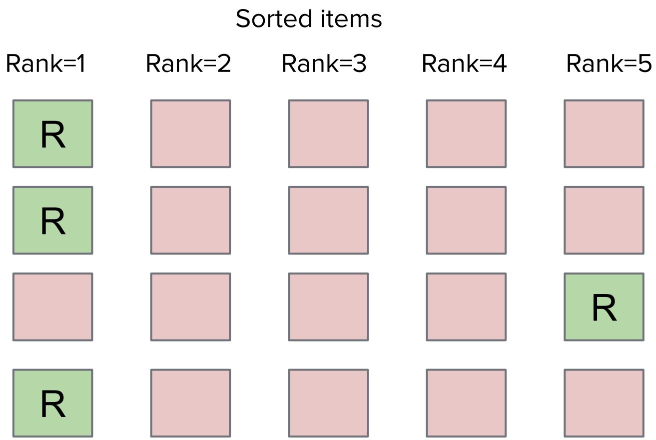
- Music: Free Music Archive

dataset	clips	genres	length	size	
			[s]	[GiB]	#days
small	8,000	8	30	7.4	2.8
medium	25,000	16	30	23	8.7
large	106,574	161	30	98	37
full	106,574	161	278	917	343

Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. Fma: A dataset for music analysis. arXiv preprint arXiv:1612.01840, 2016.

03 Evaluation

- **Coarse search:** Audio file with maximum matching segments
- **Fine-grained search:**
  - start - end matching times
  - find candidate sequence of segments
  - edit distance
- **Metric:**
  - $\text{recall@top-k} = \frac{\text{n hits @ top-k}}{\text{n hits @ top-k + m miss @ top-k}}$
  - $\text{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank(q_i)}$



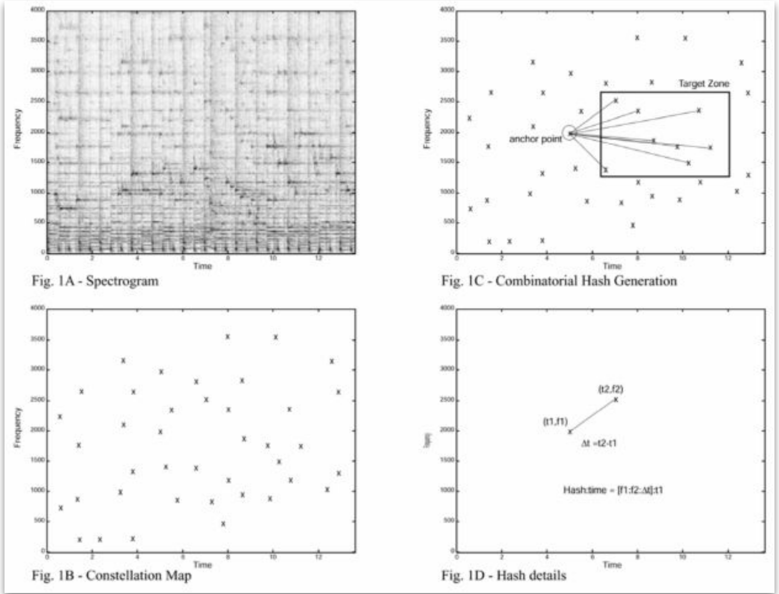
Coding time!

4

Literature Review

04 Shazam

- Identifies frequencies of peak intensity
- Generates pairs of anchor and target peaks and their corresponding time difference.
- Hashes each pair into a hash table.
- Hash table lookup for query matching.

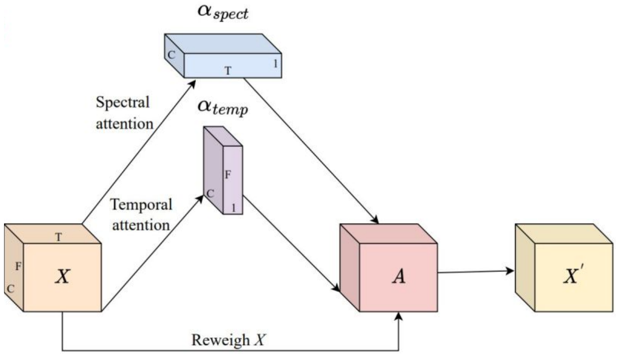


Avery Wang et al. An industrial strength audio search algorithm. In ISMIR, volume 2003, pp. 7–13, 2003.

04 ML approaches

- Enhances interesting spatial patches by assigning more weight to time indices and frequency bands containing salient patches

$$a^{temp} = \mathrm{softmax}(X^T W_{temp})$$
$$a^{spect} = \mathrm{softmax}(X^T W_{spect})$$
$$A = a^{temp} \otimes a^{spect} \times s$$
$$X' = A \odot X$$

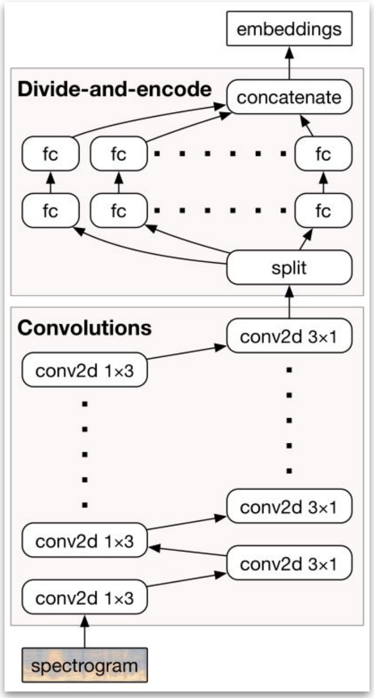


Layer	Input size	Output size
Encoder:		
CNN layer	1×64×96	32×64×96
ResBlock1	32×64×96	32×64×96
ResBlock2	32×64×96	64×32×48
...		
ResBlock6	512×4×6	1024×2×3
Flatten		6144
Projection Head:	$d * i$	$d * o$
Conv1D + ELU	128×48	128×32
Conv1D	128×32	128×1

Singh, Anup, Kris Demuynck, and Vipul Arora. "Attention-based audio embeddings for query-by-example." *arXiv preprint arXiv:2210.08624* (2022).

04 ML approaches

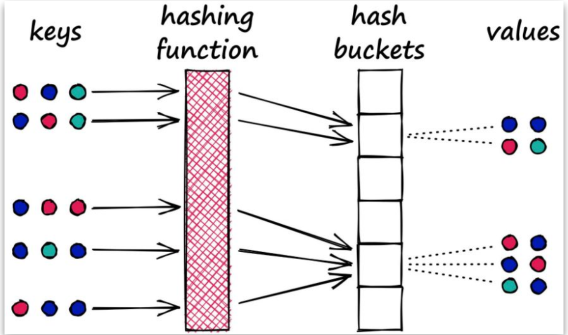
- CNN architecture with spatially separable convolutions layers.
- Divide-and-encode: splits embedding into chunks
- Triplet loss



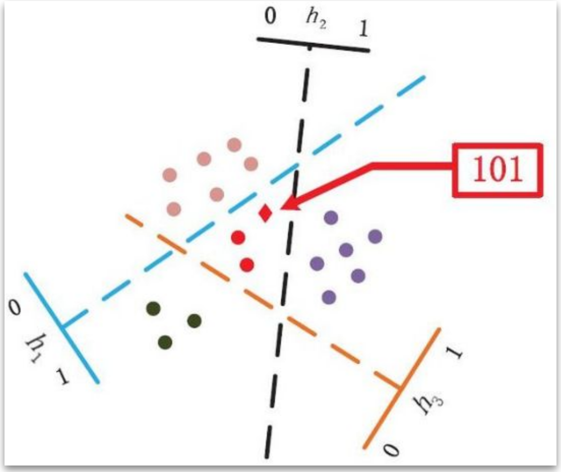
Gfeller, Beat, et al. "Now playing: Continuous low-power music recognition." *arXiv preprint arXiv:1711.10958* (2017).

04 LSH

- Map similar samples to same hash code
- K random hyperplanes: $h_1, h_2, \ldots h_K \Rightarrow$ 2^K disjoint partitions of space.
- K-bit hash code for sample **a**: Step $[a^T h_1\ a^T h_2 \ldots\ a^T h_K]$
- Exact comparison of **a** with points mapped to same hash bucket - May miss near neighbors
- Repeat L times w.r.t different set of K hyperplanes.
- Distance computation with candidates mapped to same hash code as **a** across L tables.



Source: https://www.pinecone.io/learn/series/faiss/locality-sensitive-hashing/

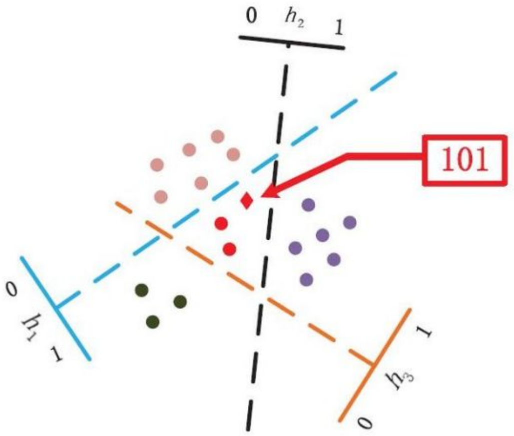


Source: https://www.researchgate.net/publication/314300245_Feature_Matching_of_Multi-view_3D_Models_Based_on_Hash_Binary_Encoding

Lv, Qin, et al. "Multi-probe LSH: efficient indexing for high-dimensional similarity search." Proceedings of the 33rd international conference on Very large data bases. 2007.

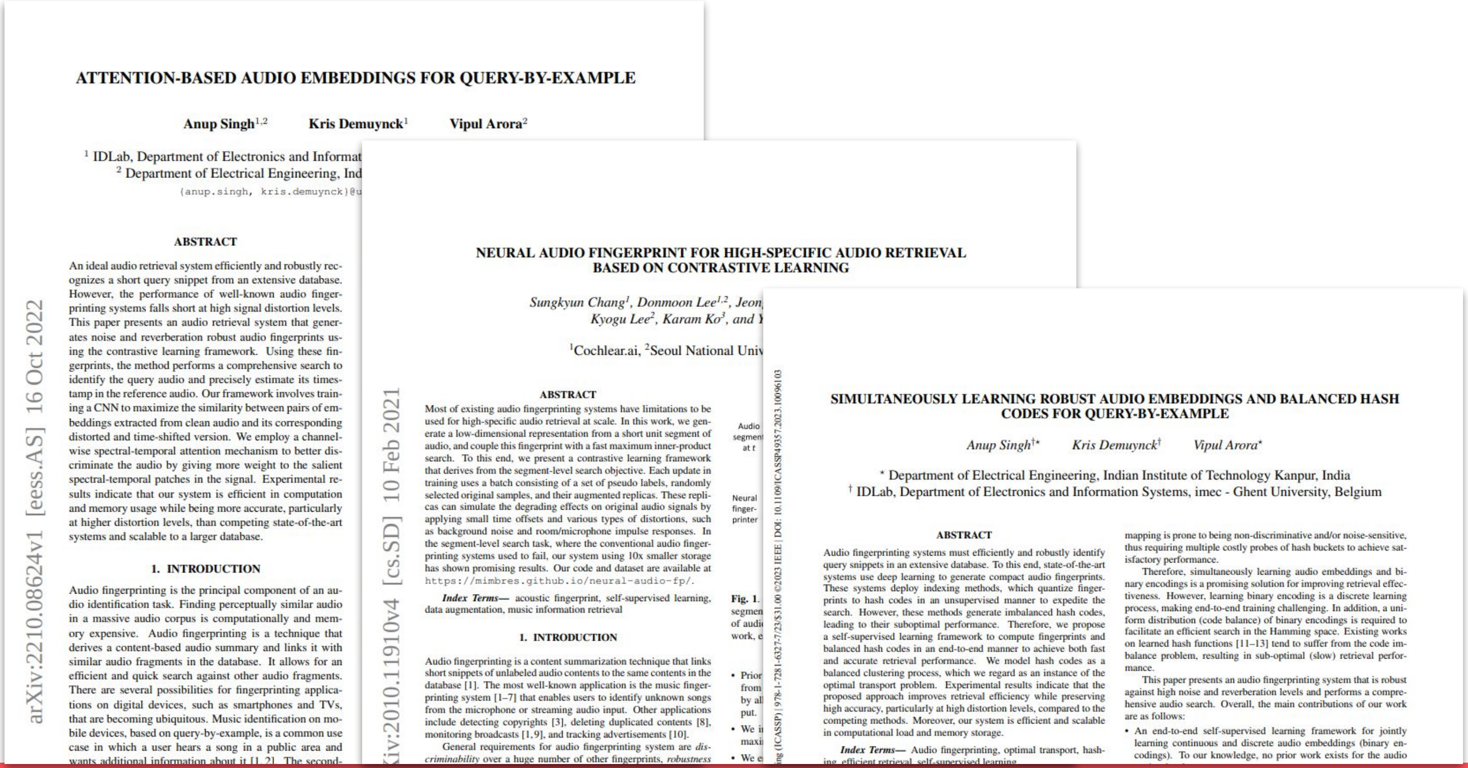
04 LSH

- **Computational Cost:**
  - N points, D dimensional, K hyperplanes
  - DK: generate K-bit hash code ⇒ Mapping sample to a hash bucket. Cost of dot product of a sample with K hyperplanes.
  - Assume on average, each bucket contains: N/2^K
  - Exact comparison cost: DN/2^K
  - Repeat everything L times (no. of hash tables)
  - Cost: LDK + LDN/2^K → O(log N), if K = log N



Lv, Qin, et al. "Multi-probe LSH: efficient indexing for high-dimensional similarity search." Proceedings of the 33rd international conference on Very large data bases. 2007.

04 Recent Works



arXiv:2210.08624v1 [eess.AS] 16 Oct 2022

ATTENTION-BASED AUDIO EMBEDDINGS FOR QUERY-BY-EXAMPLE

Anup Singh[1,2] Kris Demuynck[1] Vipul Arora[2]

[1] IDLab, Department of Electronics and Informat
[2] Department of Electrical Engineering, Ind
{anup.singh, kris.demuynck}@u

ABSTRACT

An ideal audio retrieval system efficiently and robustly recognizes a short query snippet from an extensive database. However, the performance of well-known audio fingerprinting systems falls short at high signal distortion levels. This paper presents an audio retrieval system that generates noise and reverberation robust audio fingerprints using the contrastive learning framework. Using these fingerprints, the method performs a comprehensive search to identify the query audio and precisely estimate its timestamp in the reference audio. Our framework involves training a CNN to maximize the similarity between pairs of embeddings extracted from clean audio and its corresponding distorted and time-shifted version. We employ a channel-wise spectral-temporal attention mechanism to better discriminate the audio by giving more weight to the salient spectral-temporal patches in the signal. Experimental results indicate that our system is efficient in computation and memory usage while being more accurate, particularly at higher distortion levels, than competing state-of-the-art systems and scalable to a larger database.

1. INTRODUCTION

Audio fingerprinting is the principal component of an audio identification task. Finding perceptually similar audio in a massive audio corpus is computationally and memory expensive. Audio fingerprinting is a technique that derives a content-based audio summary and links it with similar audio fragments in the database. It allows for an efficient and quick search against other audio fragments. There are several possibilities for fingerprinting applications on digital devices, such as smartphones and TVs, that are becoming ubiquitous. Music identification on mobile devices, based on query-by-example, is a common use case in which a user hears a song in a public area and wants additional information about it [1, 2]. The second-

Xiv:2010.11910v4 [cs.SD] 10 Feb 2021

NEURAL AUDIO FINGERPRINT FOR HIGH-SPECIFIC AUDIO RETRIEVAL BASED ON CONTRASTIVE LEARNING

Sungkyun Chang[1], Donmoon Lee[1,2], Jeon
Kyogu Lee[2], Karam Ko[3], and Y

[1]Cochlear.ai, [2]Seoul National Univ

ABSTRACT

Most of existing audio fingerprinting systems have limitations to be used for high-specific audio retrieval at scale. In this work, we generate a low-dimensional representation from a short unit segment of audio, and couple this fingerprint with a fast maximum inner-product search. To this end, we present a contrastive learning framework that derives from the segment-level search objective. Each update in training uses a batch consisting of a set of pseudo labels, randomly selected original samples, and their augmented replicas. These replicas can simulate the degrading effects on original audio signals by applying small time offsets and various types of distortions, such as background noise and room/microphone impulse responses. In the segment-level search task, where the conventional audio fingerprinting systems used to fail, our system using 10x smaller storage has shown promising results. Our code and dataset are available at https://mimbres.github.io/neural-audio-fp/.

Index Terms— acoustic fingerprint, self-supervised learning, data augmentation, music information retrieval

1. INTRODUCTION

Audio fingerprinting is a content summarization technique that links short snippets of unlabeled audio contents to the same contents in the database [1]. The most well-known application is the music fingerprinting system [1–7] that enables users to identify unknown songs from the microphone or streaming audio input. Other applications include detecting copyrights [3], deleting duplicated contents [8], monitoring broadcasts [1,9], and tracking advertisements [10].

General requirements for audio fingerprinting system are *discriminability* over a huge number of other fingerprints, *robustness*

Audio segment at *t*

Neural finger-printer

Fig. 1.
segmen
of audi
work, e

• Prior
from
by al
put.
• We i
maxi
• We e

ng (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10096103

SIMULTANEOUSLY LEARNING ROBUST AUDIO EMBEDDINGS AND BALANCED HASH CODES FOR QUERY-BY-EXAMPLE

Anup Singh[†⋆] *Kris Demuynck*[†] *Vipul Arora*[⋆]

[⋆] Department of Electrical Engineering, Indian Institute of Technology Kanpur, India
[†] IDLab, Department of Electronics and Information Systems, imec - Ghent University, Belgium

ABSTRACT

Audio fingerprinting systems must efficiently and robustly identify query snippets in an extensive database. To this end, state-of-the-art systems use deep learning to generate compact audio fingerprints. These systems deploy indexing methods, which quantize fingerprints to hash codes in an unsupervised manner to expedite the search. However, these methods generate imbalanced hash codes, leading to their suboptimal performance. Therefore, we propose a self-supervised learning framework to compute fingerprints and balanced hash codes in an end-to-end manner to achieve both fast and accurate retrieval performance. We model hash codes as a balanced clustering process, which we regard as an instance of the optimal transport problem. Experimental results indicate that the proposed approach improves retrieval efficiency while preserving high accuracy, particularly at high distortion levels, compared to the competing methods. Moreover, our system is efficient and scalable in computational load and memory storage.

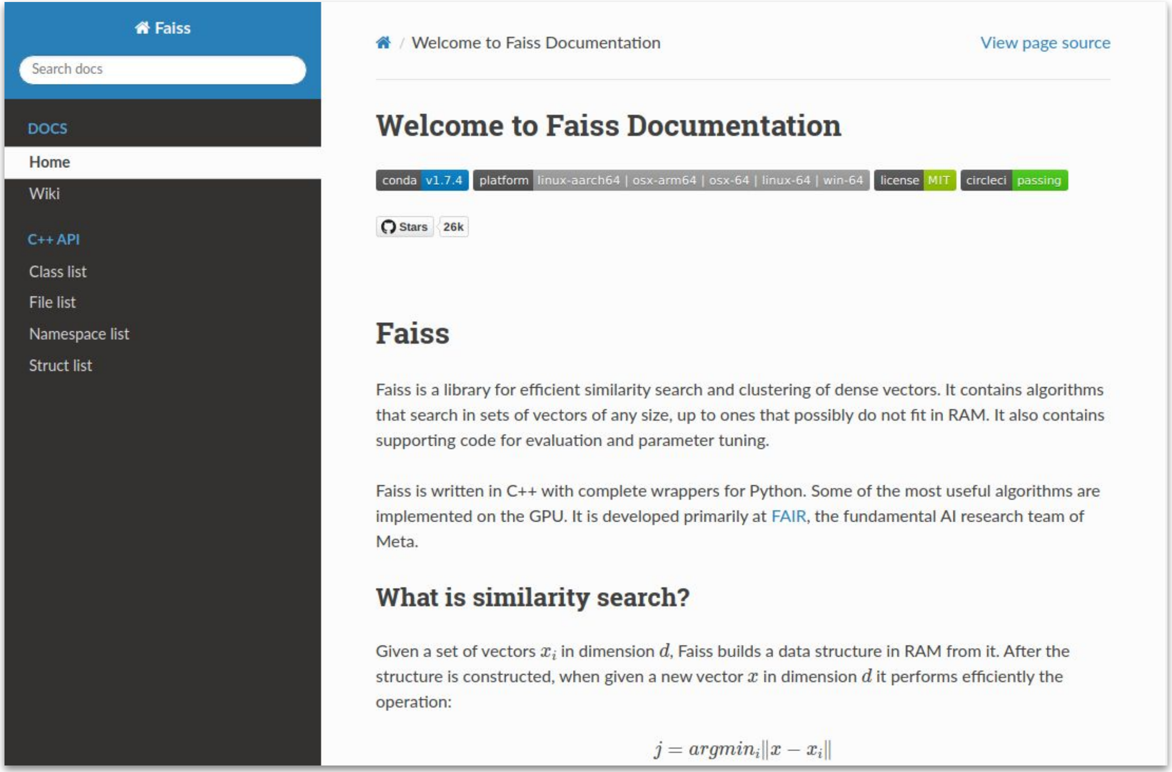
Index Terms— Audio fingerprinting, optimal transport, hashing, efficient retrieval, self-supervised learning

mapping is prone to being non-discriminative and/or noise-sensitive, thus requiring multiple costly probes of hash buckets to achieve satisfactory performance.

Therefore, simultaneously learning audio embeddings and binary encodings is a promising solution for improving retrieval effectiveness. However, learning binary encoding is a discrete learning process, making end-to-end training challenging. In addition, a uniform distribution (code balance) of binary encodings is required to facilitate an efficient search in the Hamming space. Existing works on learned hash functions [11–13] tend to suffer from the code imbalance problem, resulting in sub-optimal (slow) retrieval performance.

This paper presents an audio fingerprinting system that is robust against high noise and reverberation levels and performs a comprehensive audio search. Overall, the main contributions of our work are as follows:

• An end-to-end self-supervised learning framework for jointly learning continuous and discrete audio embeddings (binary encodings). To our knowledge, no prior work exists for the audio



Website: https://faiss.ai/

For more information ...

- Contact me at:
  - Email: anup.singh@ugent.be
  - Twitter: @15_anup

Thank you!