A Semi-Supervised Deep Learning Approach to Dataset Collection for Query-by-Humming Task

Amantur Amatov [1] Dmitry Lamanov [2] Maksim Titov [2] Ivan Vovk [2] Ilya Makarov [3] Mikhail Kudinov [2]

[1]Higher School of Economics [2]Huawei Noah's Ark Lab [3]AI Center, NUST MISiS

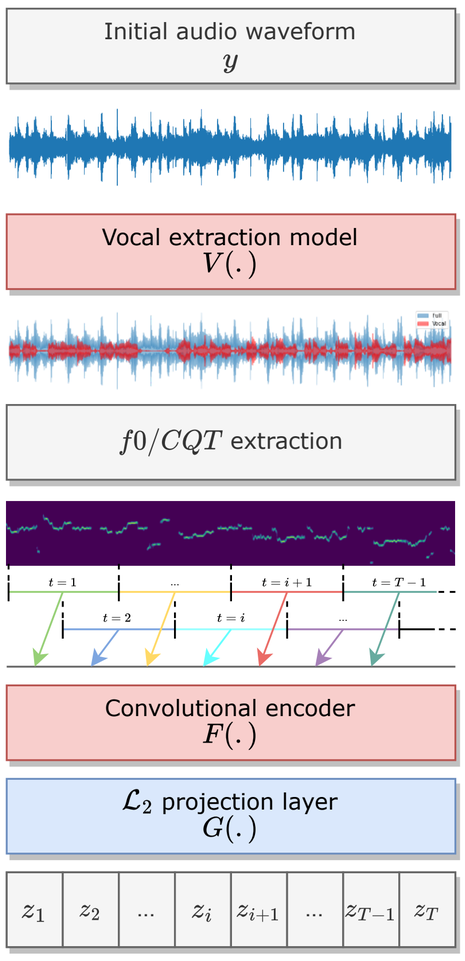
Highlights

- We present a **semi-supervised data collection and training pipeline** for the **Query-by-Humming** task, utilizing it as a specialized instance of the **Cover Song Identification** task.
- We contribute a novel dataset - **Covers and Hummings Aligned Dataset (CHAD)**, comprising **18 hours** of short music fragments paired with time-aligned hummed versions collected through crowdsourcing. Additionally, our pipeline has extended the dataset to include **over 300 hours** of music fragments paired with time-aligned cover versions.
- We demonstrate the **effectiveness** of employing cover songs to train Query-by-Humming models, resulting in competitive performance on both **benchmark datasets** and our **internal large-scale dataset**.
- The dataset download script is available on our **GitHub page**!

Backbone model

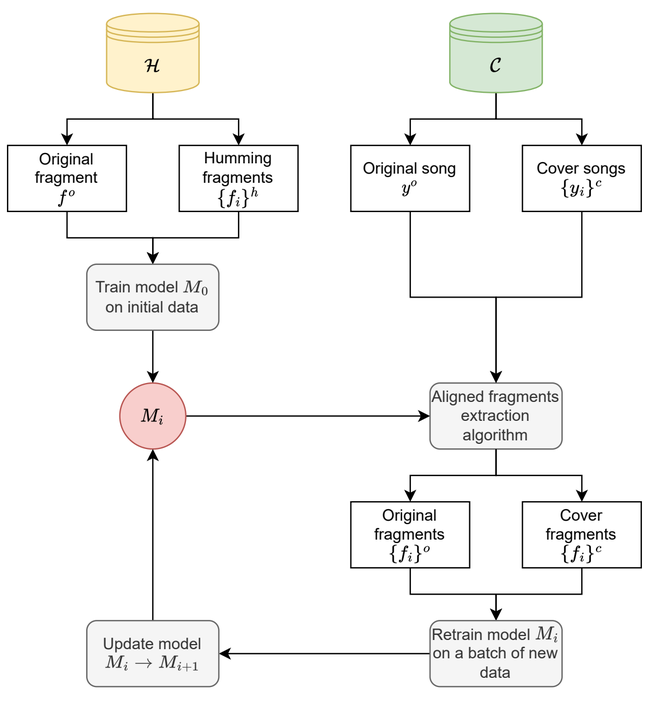
1. Use the pre-trained audio source separation model $V(.)$ to extract the **vocal part** from the signal y.
2. Extract **spectral features**, either the fundamental frequency ($f0$) using $CREPE$ [1] or the Constant-Q Transform (CQT), from the vocal part.
3. Apply a ResNet18-based [2] convolutional encoder $F(.)$.
4. Apply $\mathcal{L}_2$-normalisation layer $G(.)$.
5. Obtain output fingerprints $Z = z_{ii=1...T}$, where T is the total number of fingerprints, and each fingerprint has a dimension size of 128.
6. **Metric learning** loss function:

$$\ell = -\sum_{k=1}^{K} \sum_{z_k^i, z_k^j \in Z_k} \log \frac{\exp(\frac{sim(z_k^i, z_k^j)}{\tau})}{\sum_{z_l \notin Z_k} \exp(\frac{sim(z_k^i, z_l)}{\tau})}$$

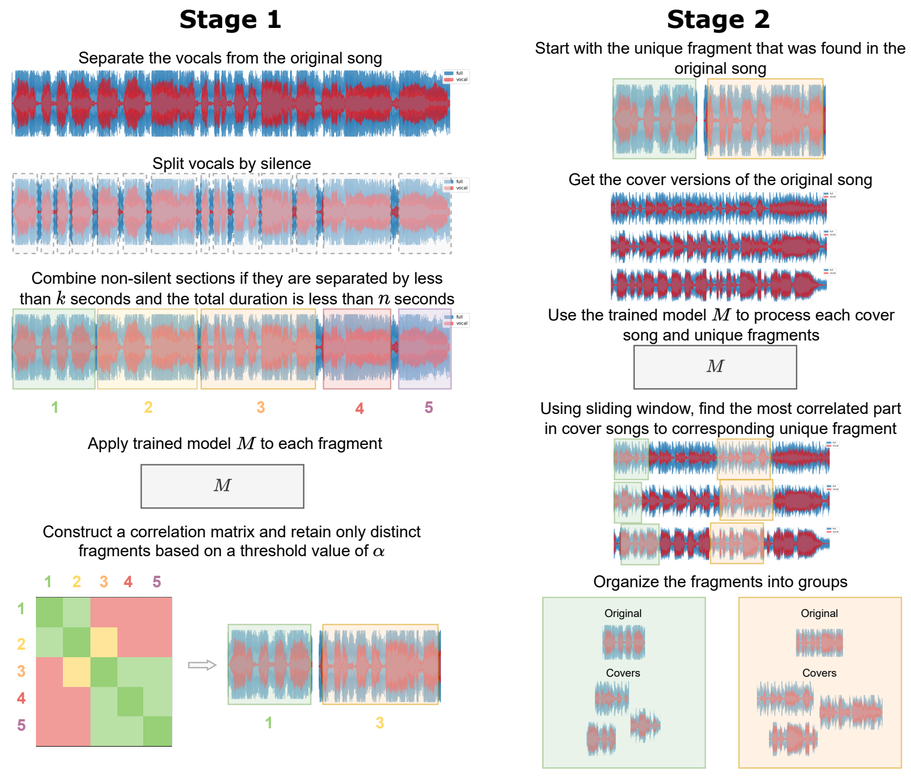


Semi-supervised pipeline

1. Train the **initial model** M_0 using hummed fragments collected via crowdsourcing.
2. Collect **groups of cover songs**, either by scraping from YouTube or using open-source datasets.
3. Using M_0, extract aligned fragments from each group of cover songs using **aligned fragments extraction algorithm**.
4. **Retrain model** on newly gathered data and repeat the process.



Aligned fragments extraction algorithm



Cover and Hummings Aligned Dataset

- **CHAD** contains **5494** original songs, **31630** cover songs, and **5164** hummings fragments.
- **81781** audio fragments with **270 hours** of singing/humming and **51 hours** of original song fragments.
- In hummings subset $\mathcal{H}$, the total duration for original fragments - **2.12 hours**, and for humming fragments - **15.83 hours**.
- In covers subset $\mathcal{C}$, the total duration for original fragments - **49.54 hours**, and cover fragments - **259.03 hours**.
- **The metadata** includes YouTube ID, title, author, cover fragment correlation values, time interval, and whether it is double-checked.

Results

- Results on benchmark datasets:

	Method	Top-10 hit rate ↑				
		Jang[3]	Thinkit	Subtask 2	Jang Real	MTG-QBH [4]
Ours	metric learning(CREPE)	0.921	0.966	0.959	0.868	0.883
	metric learning(CQT)	0.840	0.786	0.866	0.867	0.747
Stasiak [5]	$f0$-matching	0.948	0.907	0.968	-	-
ACRCloud	proprietary	**0.990**	**0.986**	**0.972**	-	-

- Results on a large-scale internal dataset of 90k songs:

Partition	Model	Top-n hit rate↑			
		100	10	5	3
$\mathcal{C}$	M_{short}	0.643	0.548	0.524	0.476
	M_{long}	0.412	0.277	0.270	0.262
	M_{fused}	0.759	0.621	0.603	0.517
$\mathcal{C}+\mathcal{H}$	M_{short}	0.659	0.595	0.571	0.484
	M_{long}	0.595	0.508	0.413	0.389
	M_{fused}	0.776	0.707	0.691	0.586

Results on humming queries.

Partition	Model	Top-n hit rate↑			
		100	10	5	3
$\mathcal{C}$	M_{fused}	0.931	0.904	0.885	0.865
$\mathcal{C}+\mathcal{H}$	M_{fused}	0.923	0.899	0.885	0.856

Results on singing queries.

Model	ANN	Reranking
M_{short}	1.41 ± 0.57	5.37 ± 0.87
M_{long}	0.52 ± 0.11	2.39 ± 0.43

Search speed.

References

[1] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," 2018.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[3] J.-S. R. Jang, "Qbsh: A corpus for designing qbsh (query by singing/humming) systems."

[4] J. Salamon, J. Serrà, and E. Gómez, "Tonal representations for music retrieval: From version identification to query-by-humming," *International Journal of Multimedia Information Retrieval, special issue on Hybrid Music Information Retrieval*, vol. 2, pp. 45–58, 03 2013.

[5] B. Stasiak, "Follow that tune – adaptive approach to dtw-based query-by-humming system," *Archives of Acoustics*, vol. 39, pp. 467 –, 01 2014.