ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10097162

SHOW ME THE INSTRUMENTS: MUSICAL INSTRUMENT RETRIEVAL FROM MIXTURE AUDIO

Kyungsu Kim[1*], *Minju Park*[1*], *Haesun Joung*[1*], *Yunkee Chae*[2], *Yeongbeom Hong*[1], *Seonghyeon Go*[1], *Kyogu Lee*[1,2,3]

[1]Department of Intelligence and Information, Seoul National University
[2]Interdisciplinary Program in Artificial Intelligence, Seoul National University
[3]Artificial Intelligence Institute, Seoul National University

ABSTRACT

As digital music production has become mainstream, the selection of appropriate virtual instruments plays a crucial role in determining the quality of music. To search the musical instrument samples or virtual instruments that make one's desired sound, music producers use their ears to listen and compare each instrument sample in their collection, which is time-consuming and inefficient. In this paper, we call this task as *Musical Instrument Retrieval* and propose a method for retrieving desired musical instruments using reference mixture audio as a query. The proposed model consists of the *Single-Instrument Encoder* and the *Multi-Instrument Encoder*, both based on convolutional neural networks. The Single-Instrument Encoder is trained to classify the instruments used in single-track audio, and we take its penultimate layer's activation as the instrument embedding. The Multi-Instrument Encoder is trained to estimate multiple instrument embeddings using the instrument embeddings computed by the Single-Instrument Encoder as a set of target embeddings. For more generalized training and realistic evaluation, we also propose a new dataset called *Nlakh*. Experimental results showed that the Single-Instrument Encoder was able to learn the mapping from the audio signal of unseen instruments to the instrument embedding space and the Multi-Instrument Encoder was able to extract multiple embeddings from the mixture audio and retrieve the desired instruments successfully. The code used for the experiment and audio samples are available at: `https://github.com/minju0821/musical_instrument_retrieval`

Index Terms— music information retrieval, musical instrument, dataset

1. INTRODUCTION

Nowadays, advances in digital music production technology enabled the musicians to explore a greater range of sonic possibilities to work with. Particularly, the development of the Digital Audio Workstation (DAW) and virtual instruments greatly expanded the space of the musical creativity [1]. As there are a large number of virtual instruments with high timbral diversity and the quality of music is highly dependent on the timbre of the instruments, selecting appropriate musical instruments plays a crucial role in digital music production.

Typical ways to retrieve proper musical instruments from a large library of instruments are listening to the audio samples of the instruments one-by-one or referring to the text description of the instruments if available. However, listening to the audio samples is time-consuming and inefficient, and the descriptions are often unavailable or insufficient to express the subtle nuance of the timbre of the musical instruments [2].

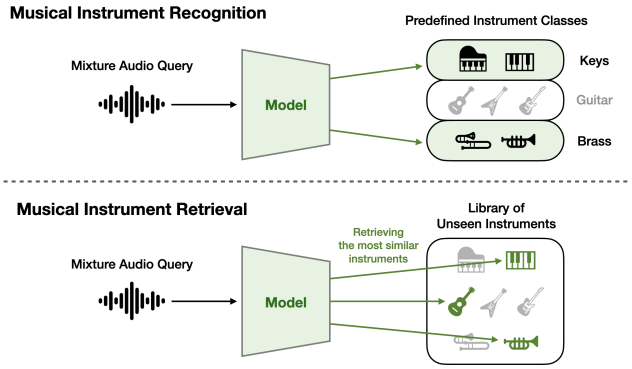


Fig. 1: Comparison between musical instrument recognition and retrieval task.

We call this task of retrieving specific desired instruments from the library of musical instruments as *Musical Instrument Retrieval*. Since musicians often refer to existing music to describe the sound they want, we propose to use reference music as a query for musical instrument retrieval. In this task, the objective is to retrieve instruments from the instrument library that are most similar to the instruments present in the given mixture audio. Note that the library contains instruments that were unseen during the training process. In our experiment, for quantitative evaluation, the instrument used for mixture audio query was always included in the library. We evaluated whether the model retrieved the exact instruments used in mixture audio query in terms of F1 score and mean Average Precision (mAP).

Musical instrument recognition is a closely related task that has been actively studied in the field of music information retrieval [3, 4, 5, 6, 7, 8]. However, existing approaches for musical instrument recognition are mostly focused on classification tasks that are limited to predicting the coarse categories of the instrument. While recent studies such as [9] and [10] have shown progress in classifying previously unseen musical instruments for fine-grained classes, these methods have limitations in recognizing multiple instruments from a mixture audio query. Comparison between the two tasks is illustrated in Fig. 1.

Our proposed method employs the Single-Instrument Encoder and the Multi-Instrument Encoder. The former is responsible for

*Equal contribution.

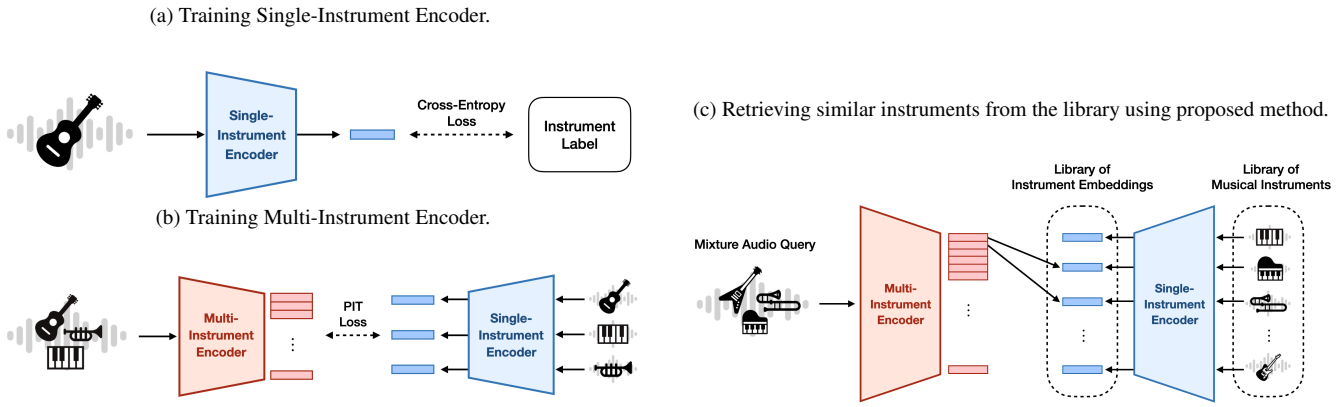


Fig. 2: The overall process of the suggested method. (a) Single-Instrument Encoder is trained to classify which instrument played the input audio. We take the penultimate layer's activation of the trained network as instrument embedding. (b) Multi-Instrument Encoder extracts multiple instrument embeddings from the mixture audio. The Single-Instrument Encoder provides the set of target embeddings. (c) At inference time, we first extract the instrument embeddings of each instrument in the instrument library for a single time. Then we extract the multiple embeddings from the mixture audio query and retrieve the most similar instruments from the instrument library.

extracting instrument embeddings from single-track audio, while the latter is trained to extract multiple instrument embeddings from mixture audio by using the embeddings extracted by the Single-Instrument Encoder as the target. Since we estimate the set of embeddings, which is permutation-invariant, we use permutation invariant training (PIT) [11] scheme for Multi-Instrument Encoder training.

To meet the dataset requirements for training and evaluating a general instrument encoder, we introduce a new dataset called *Nlakh*(pronounced as en-läk), which combines the NSynth dataset [12] and the Lakh dataset [13, 14]. The Nlakh dataset enables the model to extract embeddings robustly to instrument combinations and performance.

Our experimental results show that the Single-Instrument Encoder successfully maps different audio samples of the same instruments into close embeddings, while the Multi-Instrument Encoder can separate mixture audio at the embedding level and successfully retrieve desired instruments from the library.

2. RELATED WORKS

Studies on retrieving musical instruments or extracting instrument embeddings are still in its early stages. Recently, [15] has trained and evaluated a model for extracting instrument embedding from a music signal by adopting the framework of speaker verification task, but the model was limited to extracting an embedding from single-sourced audio. Musical instrument retrieval methods with audio query have also been studied recently, but mostly focusing on retrieving drum samples. [16] adopts deep convolutional auto-encoders to retrieve drum samples by using vocal imitations as the query. Furthermore, [17] conducts deep metric learning to extract the drum embeddings from a mixture audio as the query. In this paper, we expand this approach for retrieving multi-pitched instruments. [9] [10] [18]

3. METHOD

The proposed model consists of the Single-Instrument Encoder and the Multi-Instrument Encoder. The Single-Instrument Encoder extracts an instrument embedding from a single-track audio of the instrument. Using the instrument embeddings computed by the Single-Instrument Encoder as a set of target embeddings, the Multi-Instrument Encoder is trained to estimate the multiple instrument embeddings. As we estimate the set of embeddings, which is permutation-invariant, PIT scheme [11] was used for training. The overall framework of the proposed model is depicted in Fig. 2.

3.1. Single-Instrument Encoder

In order to extract an instrument embedding from single-track audio with the Single-Instrument Encoder, we trained a network performing classification to match the audio samples with their instrument labels. We used the network's penultimate layer's activation as the instrument embedding, which is a 1024-dimensional vector. For an instrument i_k, the Single-Instrument Encoder f extracts the embedding of the instrument i_k as $f(x_{i_k})$, where x_{i_k} is the single-track audio of the instrument i_k.

3.2. Multi-Instrument Encoder

The Multi-Instrument Encoder g aims to estimate the embeddings of a set of instruments $I = \{i_1, i_2, ..., i_N\}$ given a mixture audio $m = \sum_{i \in I} x_i$. We implemented few-shot learning using external memory [18] of target embeddings, which are the outputs of the Single-Instrument Encoder. We designed the Multi-Instrument Encoder to output M possible embeddings, where M was set as the maximum number of instruments in a mixture audio in the training set.

The Multi-Instrument Encoder g is trained to minimize the cosine embedding loss between the optimal permutation of the set of output embeddings $G = \{g(m)_{1,:}, g(m)_{2,:}, ..., g(m)_{M,:}\}$ and the set of target embeddings $F = \{f(x_1), f(x_2), ..., f(x_N)\}$. To compensate for the difference in the number of embeddings and the indeterminacy of the instrument order, we used the idea of permutation invariant training to compute the loss function [11]. The minimized

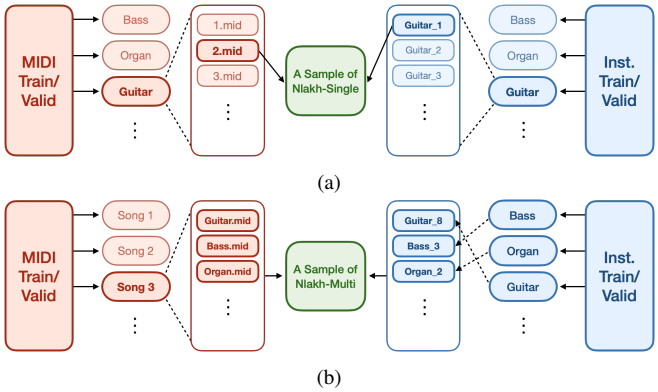


Fig. 3: The process of rendering a sample of (a) Nlakh-single and (b) Nlakh-multi

loss function is described as follows:

$$\mathcal{L} = \min_{\pi} \sum_{n=1}^{N} \left(1 - \cos\theta_{\pi(n),n}\right)$$

$$\cos\theta_{\pi(n),n} = \frac{g(m)_{\pi(n),:} \cdot f(x_n)}{||g(m)_{\pi(n),:}|| \cdot ||f(x_n)||}$$

where $\pi : \{1, 2, \ldots, N\} \mapsto \{1, 2, \ldots, M\}$ is an injective function.

To minimize the computational cost of finding the optimal permutation, we applied the optimal permutation invariant training method that utilizes the Hungarian algorithm [19, 20].

3.3. Inference

To use the trained encoders for retrieval task, for each instrument l_k in instrument library $L = \{l_1, l_2, l_3, ..., l_K\}$, we extract the instrument embedding $f(x_{l_k})$ to construct the embedding library $E = \{f(x_{l_1}), f(x_{l_2}), ..., f(x_{l_K})\}$ using the trained Single-Instrument Encoder. Given the mixture audio query m, we extract output embeddings $\{g(m)_{1,:}, ..., g(m)_{M,:}\}$ using the trained Multi-Instrument Encoder. Then we calculate the cosine similarity $\cos\phi_{j,k}$ as follows.

$$\cos\phi_{j,k} = \frac{g(m)_{j,:} \cdot f(x_{l_k})}{||g(m)_{j,:}|| \cdot ||f(x_{l_k})||}$$

For each output embedding $g(m)_{j,:}$, we pick the instrument l_k whose cosine similarity $\cos\phi_{j,k}$ is the largest among other instruments in L. Therefore, the set of retrieved instruments R given mixture audio query m can be formulated as follows.

$$R = \{l_{k'} | k' \in \{\underset{k}{\operatorname{argmax}} \cos\phi_{j,k}\}_{j=1}^{M}\}$$

Note that more than two output embeddings may be assigned to the same instrument. Therefore, the size of a set R may be smaller than M.

4. THE NLAKH DATASET

To train and evaluate the proposed model, the dataset should have a large number of different instruments. Also, the dataset should contain the ensembles of different instruments to enable the model to extract instrument embeddings robustly to instrument combinations and performance. However, no existing dataset fully met these requirements. Therefore, we propose a new dataset called *Nlakh* that combines the NSynth dataset, which provides a large number of instruments, and the Lakh dataset, which provides multi-track MIDI data.

Dataset	Size (Hours)	Number of Instruments (Categories)	Stem Availability
Nlakh-single (ours)	1,397	1,006	✓
Nlakh-multi (ours)	153	1,006	✓
Slakh [21]	145	158	✓
MUSDB18 [22]	10	(5)	✓
MedleyDB [23]	7	(80)	✓
OpenMIC [24]	56	(20)	-
IRMAS [25]	6	(11)	-

Table 1: Comparison with other datasets.

Nlakh consists of *Nlakh-single* that contains single-track audio and *Nlakh-multi* that contains mixture audio with separate tracks (stem) of each instrument. To make Nlakh-single, we first separated each MIDI track of the Lakh dataset and categorized the tracks by their instrument family (bass, organ, guitar, etc.) according to the MIDI program number. Then for each instrument of NSynth, we randomly selected a five-second-long excerpt from MIDI tracks in the corresponding instrument family. For example, if the selected instrument's family is the guitar, only the MIDI files in the guitar category are used for rendering. We rendered 1,000 samples for each instrument. In total, there are 1,006,000 samples in Nlakh-single. Nlakh-single is split into train/valid set following the instrument split of NSynth (953/53).

To make Nlakh-multi, we first find a five-second-long multi-track MIDI section containing at least two valid tracks in which at least three notes are played. Likewise in Nlakh-single, we randomly selected instruments for rendering the multi-track MIDI excerpt within the corresponding instrument family. The Nlakh-multi has 100,000 samples for the training set and 10,000 samples for the validation set. The overall process of making the dataset is illustrated in Fig. 3.

Among other multi-track music datasets that contains audio data, to the best of our knowledge, Nlakh has the largest number of instruments and the largest amount of data at the same time (Table 1). In addition to the rendered audio dataset, we also provide a codebase to generate our dataset, so one can use it to render more samples.

5. EXPERIMENTS

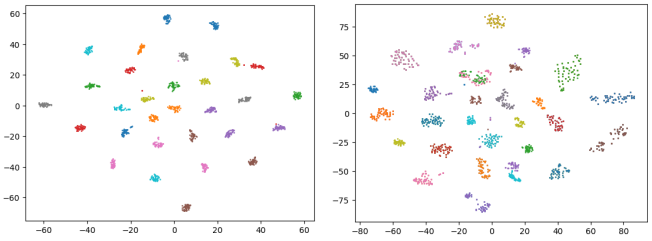
5.1. Single-Instrument Encoder

We used the convolutional neural network architecture that was used in [3] for the instrument recognition task as the backbone network of the Single-Instrument Encoder, using mel-spectrogram of the audio as the input. We used Adam optimizer with a learning rate of 0.001, and set batch size as 32.

To evaluate the Single-Instrument Encoder, we adopted the method proposed by [15], which used automatic speaker verification evaluation methodologies for evaluating the instrument embeddings. We first extract the embeddings of five different samples of the target instrument by using the trained Single-Instrument Encoder. The average of those embeddings is used as enrollment embedding. We

Table 2: Performance of the Multi-Instrument Encoder. Small/Large indicates the size of the model. Nlakh/Random indicates which dataset is used for training.

Model	Family		Instrument			
	F1		F1		mAP	
	macro	weighted	macro	weighted	macro	weighted
Chance	0.343	0.437	0.065	0.077	-	-
Small-Nlakh	0.626	0.723	0.482	0.524	0.553	0.597
Large-Nlakh	0.640	0.728	0.533	0.578	0.635	0.666
Small-Random	0.691	0.697	0.528	0.543	0.598	0.615
Large-Random	**0.814**	**0.817**	**0.694**	**0.712**	**0.752**	**0.760**



(a) (b)

Fig. 4: The t-SNE results of Single-Instrument Encoder on Nlakh-single (a) training and (b) validation dataset.

also make a comparison set that contains 20 embeddings from the target instrument and 20 embeddings from the other instruments. Then we compare each embedding in the comparison set with the enrollment embedding in terms of cosine similarity. We accept an instrument in the comparison set if the cosine similarity is above a certain threshold and reject otherwise. We sweep the threshold value and find a point where the false reject rate and false accept rate are equal. The error rate at this point is called the equal error rate (EER). We report the average value of EER, which is lower the better.

The average EER of our model on Nlakh-single was 0.026 while the previous work's EER on the NSynth dataset was 0.031. Note that the samples of the NSynth dataset contain only a single note, while the samples of Nlakh-single contain multiple notes. We also visualized the instrument embeddings of training set and validation set using t-distributed stochastic neighbor embedding (t-SNE) [26] in Figure 4. The results show that the Single-Instrument Encoder could cluster the instrument embeddings robustly even for the unseen instruments in the validation set.

5.2. Multi-Instrument Encoder

The Multi-Instrument Encoder extracts embeddings for each instrument in the mixture audio. In this experiment, we extracted up to nine embeddings, the maximum number of instruments in a single mixture in Nlakh-multi. We evaluated two different network architectures for the Multi-Instrument Encoder: the same architecture [3] as the Single-Instrument Encoder, and a larger convolutional neural network [27], since the task is more complex. For all cases, we used Adam optimizer with a learning rate of 10^{-3} and a batch size of 128.

During the experiment, we noticed an imbalance of the instrument distribution in Nlakh-multi, which may harm the performance of the trained network. To solve this issue, we also trained the network with randomly-mixed audio. We randomly selected a number of musical instruments between two and nine, and then randomly picked the audio samples of selected instruments from Nlakh-single. Those samples were used to mix the randomly-mixed audio and we mixed the audio on-the-fly during training.

Given mixture audio query, we retrieved the instruments as described in Section 3.3 and computed F1 score. We also calculated the F1 score with the instrument family as the basis of the evaluation. The instrument family is a coarsely categorized class of instruments, which is predefined in NSynth dataset. To calculate the mean Average Precision (mAP), we used the highest cosine similarity between the output embeddings and each embeddings in the embedding library as the similarity score.

Table 2 shows the evaluation results of the Multi-Instrument Encoder. We had three main observations from the evaluation. First, every trained network performed significantly better than the chance level in all measurements. Second, the network trained with randomly-mixed audio showed better generalization compared to the network trained with Nlakh-multi, as the F1 score of the latter starts to decrease after a certain amount of training. Third, the network using the larger convolutional neural network showed better performance.

6. CONCLUSION

In this work, we proposed a novel method for musical instrument retrieval that employs the Single-Instrument Encoder and the Multi-Instrument Encoder to extract the instrument embeddings. To train and evaluate the proposed model, we suggested the Nlakh dataset, which contains single-track audio and mixture audio from a large number of different musical instruments. The evaluation result showed that the Single-Instrument Encoder was able to learn the mapping from the audio signal of unseen instruments to the instrument embedding space, and the Multi-Instrument Encoder was able to extract multiple embeddings from the mixture audio and retrieve the desired instruments successfully.

7. ACKNOWLEDGEMENT

We would like to thank Changbin Jeon, Sungho Lee, and Junghyun Koo for helpful conversations and advices about this work. This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2022-0-00320, Artificial intelligence research about cross-modal dialogue modeling for one-on-one multi-modal interactions, 1/2) and Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2022 (No.R2022020066, 1/2).

8. REFERENCES

[1] Daniel A Walzer, "Independent music production: how individuality, technology and creative entrepreneurship influence contemporary music industry practices," *Creative Industries Journal*, vol. 10, no. 1, pp. 21–39, 2017.

[2] Peter Knees, Kristina Andersen, Sergi Jordà, Michael Hlatky, Günter Geiger, Wulf Gaebele, and Roman Kaurson, "Giantsteps-progress towards developing intelligent and collaborative interfaces for music production and performance," in *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2015, pp. 1–4.

[3] Yoonchang Han, Jaehun Kim, and Kyogu Lee, "Deep convolutional neural networks for predominant instrument recognition in polyphonic music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 208–221, 2016.

[4] Kleanthis Avramidis, Agelos Kratimenos, Christos Garoufis, Athanasia Zlatintsi, and Petros Maragos, "Deep convolutional and recurrent networks for polyphonic instrument classification from monophonic raw audio waveforms," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3010–3014.

[5] Agelos Kratimenos, Kleanthis Avramidis, Christos Garoufis, Athanasia Zlatintsi, and Petros Maragos, "Augmentation methods on monophonic audio for instrument classification in polyphonic music," in *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 156–160.

[6] Peter Li, Jiyuan Qian, and Tian Wang, "Automatic instrument recognition in polyphonic music using convolutional neural networks," *arXiv preprint arXiv:1511.05520*, 2015.

[7] Vincent Lostanlen and Carmine-Emanuele Cella, "Deep convolutional networks on the pitch spiral for musical instrument recognition," *arXiv preprint arXiv:1605.06644*, 2016.

[8] Kin Wai Cheuk, Keunwoo Choi, Qiuqiang Kong, Bochen Li, Minz Won, Amy Hung, Ju-Chiang Wang, and Dorien Herremans, "Jointist: Joint learning for multi-instrument transcription and its applications," *arXiv preprint arXiv:2206.10805*, 2022.

[9] Yu Wang, Justin Salamon, Mark Cartwright, Nicholas J Bryan, and Juan Pablo Bello, "Few-shot drum transcription in polyphonic music," *arXiv preprint arXiv:2008.02791*, 2020.

[10] Hugo Flores Garcia, Aldo Aguilar, Ethan Manilow, and Bryan Pardo, "Leveraging hierarchical structures for few-shot musical instrument recognition," *arXiv preprint arXiv:2107.07029*, 2021.

[11] Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 241–245.

[12] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Douglas Eck, Karen Simonyan, and Mohammad Norouzi, "Neural audio synthesis of musical notes with wavenet autoencoders," 2017.

[13] Colin Raffel, *Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching*, Columbia University, 2016.

[14] "The lakh midi dataset v0.1," `https://colinraffel.com/projects/lmd/#license`, Accessed: 2022-10-18.

[15] Xuan Shi, Erica Cooper, and Junichi Yamagishi, "Use of speaker recognition approaches for learning and evaluating embedding representations of musical instrument sounds," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 367–377, 2022.

[16] Adib Mehrabi, Keunwoo Choi, Simon Dixon, and Mark Sandler, "Similarity measures for vocal-based drum sample retrieval using deep convolutional auto-encoders," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 356–360.

[17] Wonil Kim and Juhan Nam, "Drum sample retrieval from mixed audio via a joint embedding space of mixed and single audio samples," in *Audio Engineering Society Convention 149*, Oct 2020.

[18] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM computing surveys (csur)*, vol. 53, no. 3, pp. 1–34, 2020.

[19] Shaked Dovrat, Eliya Nachmani, and Lior Wolf, "Many-speakers single channel speech separation with optimal permutation training," *arXiv preprint arXiv:2104.08955*, 2021.

[20] Harold W Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[21] Ethan Manilow, Gordon Wichern, Prem Seetharaman, and Jonathan Le Roux, "Cutting music source separation some slakh: A dataset to study the impact of training data quality and quantity," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019, pp. 45–49.

[22] Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner, "Musdb18-a corpus for music separation," 2017.

[23] Rachel M Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Pablo Bello, "Medleydb: A multitrack dataset for annotation-intensive mir research.," in *ISMIR*, 2014, vol. 14, pp. 155–160.

[24] Eric Humphrey, Simon Durand, and Brian McFee, "Openmic-2018: An open data-set for multiple instrument recognition.," in *ISMIR*, 2018, pp. 438–444.

[25] Juan J Bosch, Jordi Janer, Ferdinand Fuhrmann, and Perfecto Herrera, "A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals.," in *ISMIR*. Citeseer, 2012, pp. 559–564.

[26] Laurens Van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[27] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11976–11986.