Music Source Separation using an AutoEncoder

Francesco Brigante 1987197, Giorgia Barboni 1885285, Anja Stanic 2190471, Murad Huseynov 2181584

Abstract and Introduction

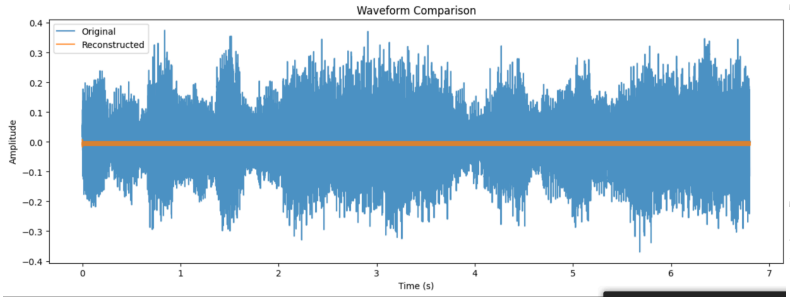
Music Source Separation (MSS) is a technique based on separating the single instrument tracks of a song. Current state-of-the-art approaches mainly rely on ground truth labels (isolated instrument tracks), which are provided only for a few instruments due to copyright issues.
Our proposed approach is based on an AutoEncoder that will learn how to reconstruct the initial song, to later apply clustering algorithms to its bottleneck layer, resulting in the separation of features relative to different instruments

Models, Dataset and Approach

We first chose a pretrained and lightweight model, Open-Unmix [1], which has been trained on MUSDB18 dataset (containing 150 songs), to perform an initial separation into the categories *drums*, *bass*, *vocals*, *others*. Our aim was to separate the *others* category even further and the main challenge was the unknown number of instruments in it.
We chose to use SEANet EnCodec, a state-of-the-art autoencoder for signal processing and we trained it on the *others* dataset, splitting it into training (130 songs) and test (20 songs)

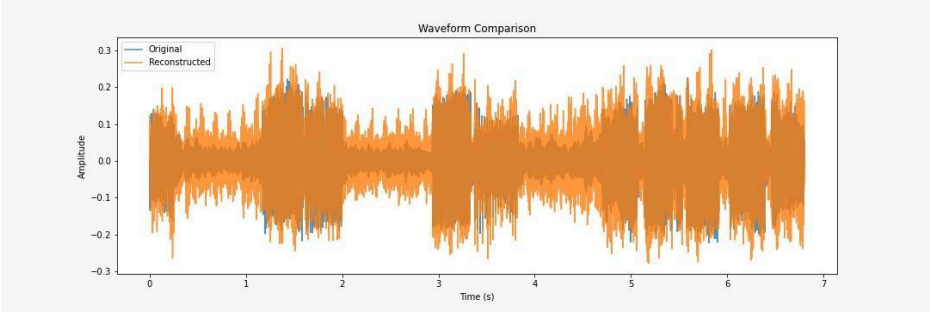
AutoEncoder Results

The baseline SEANet EnCodec architecture didn't have a loss function, so we had to choose a specific one for our task in order to train the model



The initial loss was based on computing the MSE between the original audio and its reconstruction, both in the time domain and frequency domain. This led to poor results as the latent space was collapsing and so the model wasn't learning.
Our intuition was to add a KL Divergence term to the loss function, in order to make the model generalize better and also have a well structured latent space, which follows in this case a Gaussian distribution. That is because our assumption is that the latent space is so well structured that features of similar instruments are closer, while different ones are more distant, and so we needed a constraint to enforce this assumption.

Adding this term resulted in great improvements (model trained for 24 epochs with final loss 0.1822):



Mean Square Error	Signal to Noise Ratio	Peak SNR	Cosine similarity	Spectral MSE
0.0024	1.7897 dB (5.32 dB for Open-Unmix)	26.447 dB	0.7734	13.6975

As we can see the results are pretty good, indicating that the model has learned. However there's still some noise and Spectral MSE is really high suggesting that the model is struggling in the frequency domain. Some adjustments could be trying more combinations of parameters for the spectral loss, adding a decay term to KL divergence, adding a perceptual loss to compare differences in high level features

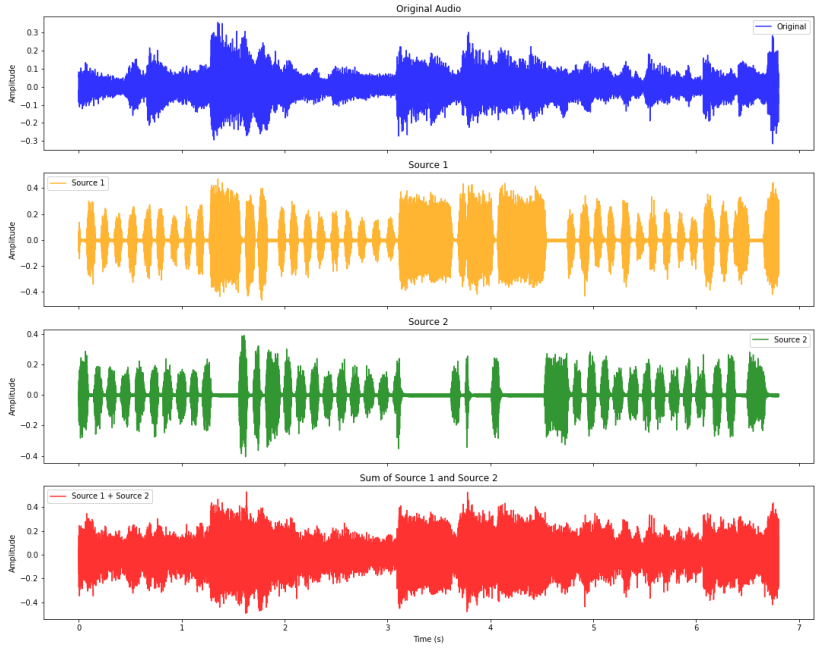
Clustering Results

We performed clustering algorithms (k-means, agglomerative, or DBSCAN) on the latent embeddings to assign frames to candidate instrument clusters, then decode each cluster separately to yield additional separated sources. Because MUSDB18 does not provide isolated references for others, we assembled a small test set derived from BASS-dB Multitrack recordings, combining stems of several instruments into a single synthetic others mix. We evaluate performance via:

- **Reconstruction Error (MSE):** the difference between the original mixture and the sum of reconstructed clusters.
- **Cluster Entropy:** a measure of how evenly frames are distributed across different clusters (lower is better)
- **Sparsity and Energy Distribution:** an inspection of energy in each cluster to check actual separation quality.

Method	# Clusters	MSE	C1 Entropy	C1 Sparsity	C1 Energy (%)	C2 Entropy	C2 Sparsity	C2 Energy (%)	Sum of Ratios
Agglomerative	2	0.0164	6.4309	0.7734	52.39	6.6802	0.7420	13.15	0.655
K-Means	2	0.0184	6.5551	0.7595	18.88	6.4346	0.7729	46.29	0.652

DBSCAN	1	0.00775	6.3847	0.7791	67.42	N/A	N/A	N/A	0.674



Conclusions and Future Works

Having a smooth, separable and well structured latent space is essential to perform Music Source Separation, which is a complex task (especially without ground truths) and requires trying different techniques and experimenting with architectures and loss functions to ensure no loss of information during encoding.
An approach based on Autoencoders for compressing audio into feature spaces seems promising and holds potential for future developments.

Some future steps could be to train the model for more time: loss was decreasing at each epoch, trying more combinations of loss functions that emphasize bottleneck structure and separability, and also experiment with different clustering algorithms or losses (for example contrastive loss)

Members Roles:

Francesco Brigante, Anja Stanic: implementing Open-Unmix, generating the dataset, implementing, training and testing SEANet EnCodec
Giorgia Barboni, Murad Huseynov: implementing and testing different clustering algorithms

References

Open-Unmix: A reference implementation for music source separation
Stöter, F.-R., Uhlich, S., Liutkus, A., & Mitsufuji, Y. (2019)
https://github.com/sigsep/open-unmix-pytorch

High Fidelity Neural Audio Compression
A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, *arXiv preprint*, arXiv:2210.13438 (2022)
https://github.com/facebookresearch/encodec

Unsupervised Blind Source Separation with Variational Auto-Encoders
Julian Neri, Roland Badeau, Philippe Depalle 2021

Unsupervised Harmonic Sound Source Separation with Spectral Clustering
Lin, Y., Wang, L., Zhang, L., & Deng, Z. (2020).

Model Selection for Deep Audio Source Separation via Clustering Analysis
Liu, A., Seetharaman, P., & Pardo, B. (2020).

Blind source separation from single channel audio recording using ICA algorithms
Piedras, J. S. C., & Orjuela-Cañón, D. (2014).

Polyphonic Instrument Recognition Using Spectral Clustering
Martins, L. G., Tzanetakis, G., & Lagrange, M. (2007).

Kernel Additive Models for Source Separation
Liutkus, A., Fitzgerald, D., & Rafii, Z. (2014).

Deep clustering: Discriminative embeddings for segmentation and separation
Hershey, J. R., Chen, Z., Roux, J. L., & Watanabe, S. (2016).

Improving music source separation based on deep neural networks through data augmentation and network blending
Uhlich, S. (2017).