MDX
Music Demixing Workshop

Danna-Sep: Unite to separate them all

Chin-Yun Yu[1] **and Kin-Wai Chuek**[2]

1 Independent Researcher **2** Information Systems and Technology Design, Singapore University of Technology and Design

In partnership with

ISMIR2021

Abstract

Deep learning-based music source separation has gained a lot of interest in the last decades. Most of the existing methods operate with either spectrograms or waveforms. Spectrogram-based models learn suitable masks for separating magnitude spectrogram into different sources, and waveform-based models directly generate waveforms of individual sources. The two types of models have complementary strengths; the former is superior given harmonic sources such as vocals, while the latter demonstrates better results for percussion and bass instruments. In this work, we improved upon the state-of-the-art (SoTA) models and successfully combined the best of both worlds. The backbones of the proposed framework, dubbed Danna-Sep[1], are two spectrogram-based models including a modified X-UMX and U-Net, and an enhanced Demucs as the waveform-based model. Given an input of mixture, we linearly combined respective outputs from the three models to obtain the final result. We showed in the experiments that, despite its simplicity, Danna-Sep surpassed the SoTA models by a large margin in terms of Source-to-Distortion Ratio.

Method

Danna-Sep is a combination of three different models: X-UMX, U-Net, and Demucs. We describe the enhancements made for each model in the following subsections.

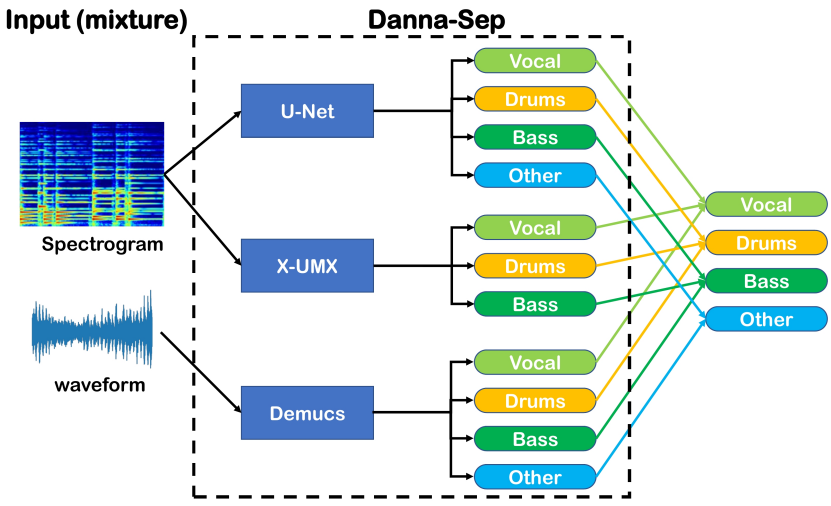

Figure 1: The schematic diagram of our proposed system.

[1] https://github.com/yoyololicon/danna-sep

X-UMX

X-UMX (Sawata et al., 2020) improved upon UMX(St "oter et al., 2019) by concatenating hidden layers of UMX to enable sharing information among all target instruments. We trained the model using the same time-domain loss as the original X-UMX, but modified the frequency-domain loss for J sources as follows:

$$\mathcal{L}_{MSE}^{J} = \sum_{j=1}^{J} \sum_{t,f} |Y_j(t,f) - \hat{Y}_j(t,f)|^2$$

where $Y_j(t,f)$ and $\hat{Y}_j(t,f)$ are ground-truth and estimated time-frequency representations for the j-th source, respectively. That is, instead of taking norm of the absolute value as in the original X-UMX, we calculated Euclidean norm in the complex domain. Also, we incorporated Multichannel Wiener Filtering (MWF)(Liutkus & Stöter, 2019) into our training pipeline in order to train our model in an end-to-end fashion. We initialized our modified X-UMX with the official pre-trained X-UMX weights[2] and continued training for approximately 70 epochs with a batch size of four.

U-Net

The encoder and decoder of our U-Net consist of six D3 Blocks (Takahashi & Mitsufuji, 2021) and we added two layers of 2D local attention (Parmar et al., 2018) layers at the bottleneck. We used the same loss function as X-UMX during training but with MWF being disabled. The approximated training time was nine days with a batch size of 16 on four Tesla V100 GPUs. We also experimented with using biaxial biLSTM along the time and frequency axes as the bottleneck layers, but it took slightly longer to train yet offered a negligible improvement.

Demucs

For Demucs (Défossez et al., 2019), we built upon the variant with 48 hidden channels, and enhanced the model by replacing the decoder with four independent decoders responsible for four respective sources. Each decoder has the same architecture as the original decoder, except for size of the hidden channel which was reduced to 24. This makes the total number of parameters comparable with the original Demucs. The training loss aggregates the L1-norm between estimated and ground-truth waveforms of the four sources. The model took approximately 10 days to train on a single RTX 3070 using mixed precision with a batch size of 16, and four steps of gradient accumulation.

Danna-Sep

In order to obtain the final output of our framework, we calculated weighted average of individual outputs from the above-mentioned models. Experiments were conducted to search for optimal weighting. The optimal weights for each source, types of input domain (T for waveforms, TF for frequency masking), and the sizes of the models are given in the following table.

	Drums	Bass	Other	Vocals	Input Domain	Size (Mb)
X-UMX	0.2	0.1	0	0.2	TF	136
U-Net	0.2	0.17	0.5	0.4	TF	61
Demucs	0.6	0.73	0.5	0.4	T	733

All models were trained on the training set of musdb18-hq (Rafii et al., 2019) using an Adam

[2] https://zenodo.org/record/4740378/files/pretrained_xumx_musdb18HQ.pth

optmizier(Kingma & Ba, 2014).

Separation performances

For a fair comparison, we trained all the models with musdb18-hq (Rafii et al., 2019) and performed the evaluation using the compressed version of the dataset (Rafii et al., 2017). One iteration of MWF was used for X-UMX and U-Net, and we didn't apply the shift trick (Défossez et al., 2019) for our enhanced Demucs. In the table below, we report the Signal-to-Distortion Ratio (SDR) (Vincent et al., 2006), calculated using *museval* (Stöter & Liutkus, 2019), attained by our modified models and the original counterparts, as well as the proposed framework.

	Drums	Bass	Other	Vocals	Avg.
X-UMX (baseline)	6.44	5.54	4.46	6.54	5.75
X-UMX (ours)	6.71	5.79	4.63	6.93	6.02
U-Net (ours)	6.43	5.35	4.67	7.05	5.87
Demucs (baseline)	6.67	6.98	4.33	6.89	6.21
Demucs (ours)	6.72	6.97	4.4	6.88	6.24
Danna-Sep	**7.2**	**7.05**	**5.2**	**7.63**	**6.77**

As can be seen from the table, our modified X-UMX gained an extra 0.27 dB on average SDR compared to the original X-UMX. The enhanced Demucs outperformed the original model by 0.03 dB of SDR, despite the fact that the shift trick was not applied. Notably, Danna-Sep surpassed both the original and enhanced Demucs by a large margin (+0.53 dB on average SDR). Altogether, the results demonstrate the efficacy of the proposed fusion method in addition to our modifications to the training scheme and architecture. The proposed framework, however, is more reliant on computing power due to the nature of model fusion, which we would like to address in furture work.

Acknowledgements

We acknowledge contributions from Sung-Lin Yeh and Yu-Te Wu, and supports from Yin-Jyun Luo and Showmin Wang during the genesis of this project.

References

Défossez, A., Usunier, N., Bottou, L., & Bach, F. (2019). Music source separation in the waveform domain. *arXiv Preprint arXiv:1911.13254*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv Preprint arXiv:1412.6980*.

Liutkus, A., & Stöter, F.-R. (2019). *Sigsep/norbert: First official norbert release* (Version v0.2.0) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.3269749

Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., & Tran, D. (2018). Image transformer. *International Conference on Machine Learning*, 4055–4064.

Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S. I., & Bittner, R. (2019). *MUSDB18-HQ - an uncompressed version of MUSDB18*. https://doi.org/10.5281/zenodo.3338373

Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S. I., & Bittner, R. (2017). *The MUSDB18 corpus for music separation*. https://doi.org/10.5281/zenodo.1117372

Sawata, R., Uhlich, S., Takahashi, S., & Mitsufuji, Y. (2020). *All for one and one for all: Improving music separation by bridging networks*. http://arxiv.org/abs/2010.04228

St "oter, F.-R., Uhlich, S., Liutkus, A., & Mitsufuji, Y. (2019). Open-unmix - a reference implementation for music source separation. *Journal of Open Source Software*. https://doi.org/10.21105/joss.01667

Stöter, F.-R., & Liutkus, A. (2019). *Museval 0.3.0* (Version v0.3.0) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.3376621

Takahashi, N., & Mitsufuji, Y. (2021). Densely connected multidilated convolutional networks for dense prediction tasks. *Proc. CVPR*.

Vincent, E., Gribonval, R., & Févotte, C. (2006). Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, *14*(4), 1462–1469.