https://doi.org/10.1007/s11760-022-02269-1

ORIGINAL PAPER

A novel U-Net with dense block for drum signal separation from polyphonic music signal mixture

E. Vinitha George[1] · V. P. Devassia[2]

Received: 16 October 2021 / Revised: 1 May 2022 / Accepted: 14 May 2022 / Published online: 16 June 2022

Abstract

Deep neural network algorithms have shown promising results for music source signal separation. Most existing methods rely on deep networks, where billions of parameters need to be trained. In this paper, we propose a novel autoencoder framework with a reduced number of parameters to separate the drum signal component from a music signal mixture. A denoising autoencoder with a U-Net architecture and direct skip connections was employed. A dense block is included in the bottleneck of the autoencoder stage. This technique was tested on both demixing secret data (DSD) and the MUSDB database. The source-to-distortion ratio (SDR) for the proposed method was at par with that of other state-of-the-art methods, whereas the number of parameters required was quite low, making it computationally more efficient. The experiment performed using the proposed method to separate drum signal yielded an average SDR of 5.71 on DSD and 6.45 on MUSDB database while using only 0.32 million parameters.

Keywords Audio source separation · Convolutional neural network · Deep learning · Denoising autoencoder · U-Net architecture

1 Introduction

Audio source separation involves the separation of the constituent audio signals from a composite audio mixture captured by an array of microphones. Audio signals of research interest in entertainment areas generally include speech and music. Music signals contain certain characteristics that differentiate them from speech and other non-musical signals because of differences in sound producing mechanism [1]. Audio source separation is applied in music editing, musical remixing, audio information retrieval, etc. The audio source separation techniques mostly depend on the time–frequency characteristics of the audio mixture, the number of sources and the spectral characteristics of sources.

Various algorithms have been developed for audio source separation, which mainly fall into two types, namely the *unsupervised* and the *supervised algorithms*. Popular unsupervised algorithms include independent component analysis (ICA) [2] and nonnegative matrix factorization (NMF) [3]. ICA finds an inverse mixing matrix that estimates the most independent source signal. In many non-musical scenarios, audio sources are uncorrelated in their behavior and have relatively less overlap in time and frequency. However, in music, sources are often strongly correlated in onset and offset times. Music sources often have strong frequency overlap too. Therefore, time and frequency overlap are significant features in the music signal mixture, in addition to the strong correlation between the sources. In such cases, the performance of ICA decreases, and hence, ICA-based approaches are not effective for music source separation.

The NMF works in the time–frequency (TF) domain. It decomposes the magnitude spectrogram of mixture signals into additive part-based decompositions, called *basis functions*. NMF-based drum source separation was performed by relying on prior information [4]. The NMF method assumed that the signal mixture is a linear combination of sources. Hence, NMF is not suitable for handling complex sources.

✉ E. Vinitha George
vinithageorge73@gmail.com

V. P. Devassia
vpdmec@yahoo.com

[1] Model Engineering College, Cochin University of Science and Technology, Kochi, Kerala, India

[2] St. Joseph's College of Engineering and Technology, Palai, Kerala, India

Supervised algorithms include support vector machines (SVM) and deep neural networks (DNN). SVMs are generally preferred for classification [5,6]. DNNs use nonlinear models trained with a large number of parameters. In general, these parameters are weights that are learned during training. The processing time required by the network increases as the number of network parameters increases. The obvious benefit of having many parameters is that more complicated functions can be represented. On the other hand, with fewer parameters, the network is flexible and the issues arising from overfitting can be prevented.

Recent deep learning models for audio source separation use either spectrogram-based or waveform-based models. Spectrogram-based models are trained to estimate masks such as binary or soft masks of the desired target source, which are used to obtain the estimates of the corresponding sources [7,8]. In contrast, waveform-based models separate sources directly in the time domain. DNN models working directly on time-domain waveforms require larger convolution kernels than spectrogram-based models because of the higher time resolution in the time-domain waveforms [8].

Waveform-based models such as Wave-U-Net [9], Meta-TasNet [10] and Demucs [11] are used for music source separation. Even though the separation results are at par with the spectrogram-based models, the number of trainable network parameters in the waveform-based models is generally higher than that of spectrogram-based models [8].

The main objective of this study was to achieve good source separation while using a smaller number of parameters. The deep learning models for source separation using spectrogram outperformed NMF models as observed by Huang et al. [12]. Therefore, a spectrogram-based model is employed in this study. The main contribution of this paper is a U-Net architecture with skip connections in which a dense block [13] is introduced to ensure reduction in training parameters.

The rest of this paper is organized as follows. The state-of-the-art spectrogram-based music signal separation techniques using neural networks are described in detail in Sect. 2. The proposed drum signal separation approach is described in Sect. 3. The experiment along with ablation study is explained in Sect. 4. The results of the investigation and comparison with other popular methods are presented in Sect. 5, followed by concluding remarks in Sect. 6.

2 Related work

A variety of spectrogram-based models have been presented in the literature for audio source separation [14–24]. Uhlich et al. [14] proposed a fully connected neural network (FNN) for instrument sound extraction. It discovers global features but does not exploit local time–frequency features. Chandna et al. [15] used a convolutional neural network (CNN) to separate vocals, drums, and other instruments from the mixture. This model has an encoding stage and decoding stage with horizontal, vertical and fully connected convolution layers to separate the vocals, drum and other instruments. The CNN proved to be faster than the FNN.

A convolutional denoising autoencoder (CDAE) is a special type of CNN that can efficiently denoise a signal. CDAE has been implemented for speech and music signal separation [16]. Grais and Plumbley [16] used CDAE for audio source separation in which the input spectrograms were compressed and later re-expanded to the size of the target spectrogram. It was successful in discovering global patterns, but local details were lost during contraction. Although the method yielded satisfactory results, some high-resolution information was lost during downsampling.

To ensure the reconstruction of finer high-resolution details, a U-Net architecture with skip connections between the encoder and decoder at the same hierarchical level was used. Initially, U-Net was proposed by Ronneberger et al. [25] to segment biomedical images. The U-Net architecture was later adopted by Jansson et al. [17] for audio source separation to separate vocals and instruments. The model uses skip connections, which allow information propagation between the encoder and decoder.

Apart from CNNs, models based on recurrent neural networks (RNNs) have also been employed for audio source separation. Liu and Yang [18] compared a convolution skip connection with a recurrent skip connection in a denoising autoencoder. Uhlich et al. [19] recommended data augmentation where random swapping of the left/right channel was performed for each instrument. The data augmentation strategy improved the effectiveness of source separation. To model longer temporal contexts, an RNN with long short-term memory (LSTM) was imparted into the network. Despite its good performance, the model requires a relatively long training time [20].

In an audio spectrogram, different patterns occur in different frequency bands [21]. To capture the local patterns, Takahashi and Mitsufuji [20] used a multi-scale multi-band densenet (MMDenseNet) to split the input into multiple bands. This model can efficiently learn both fine-grained local and global structures. The source separation was improved. MMDenseLSTM was proposed by Takahashi et al. [21] to utilize the sequence modeling capability of LSTMs in conjunction with MMDenseNet. This model outperformed the MMDenseNet model. This model improves the capability of source separation with fewer parameters.

Recently, Satya and Suyanto [22] have proposed a generative adversarial network (GAN) for music source separation. Here, the concept of the U-Net model was implemented on the generator. In [23], the sliced attention mechanism was used for music source separation. This is a recently developed

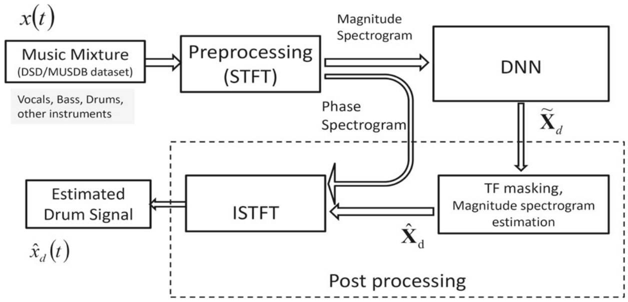

Fig. 1 General framework for signal source separation

neural network technique, which helps to learn the importance of each feature interaction from every segment of the magnitude spectrogram. The attention mechanism helped to improve the SDR in the music source separation.

3 Proposed approach

The primary goal of this study was to effectively separate the drum signal present in a polyphonic music signal. The proposed method utilizes CDAE with a U-Net architecture and a dense block.

The general framework used for the study is shown in Fig. 1. The input music mixture to be separated was a polyphonic music signal obtained from a common audio database. Typical spectrogram-based models apply short-time Fourier transform (STFT) to a mixture waveform to obtain the mixture spectrogram. These models estimate the source spectrogram from the mixture spectrogram and restore the source signal using inverse STFT (ISTFT). The time-domain signal $x(t)$ was converted to the TF domain during the preprocessing stage. The magnitude spectrogram was fed to the DNN, while the phase spectra were used later during the synthesis stage. The DNN uses the autoencoder framework to predict the target drum spectrogram, which is later subjected to postprocessing to recover the time-domain drum signal denoted by $\hat{x}_d(t)$.

3.1 Dataset

The DSD[1] and MUSDB[2] datasets [26] were used in this study. Both DSD and MUSDB are databases of professionally recorded music sources, available in stereo format with a sampling rate of 44.1 kHz. The DSD dataset contains 100 professionally recorded songs. Similarly, the MUSDB dataset has 150 songs. Each song has five wave files associated with it: the mixture signal ($x(t)$), drum signal ($x_d(t)$), the vocals, bass and other instruments. In this paper, the drum signal $x_d(t)$ is the target, whereas the vocals, bass, and other instruments are considered as the "interference."

3.2 Preprocessing

In the preprocessing stage, the actual input for the DNN is created. The stereo wave songs were converted to mono by averaging both channels. The resultant audio signals were transformed to the corresponding spectrograms using the STFT. A Hanning window of length 2048 samples was chosen with a hop size of one-fourth the window length. The complex STFT $\mathbf{X}$ of signal $x(t)$, has magnitude spectra $\mathbf{X}_{\text{Mag}}$ and phase spectra $\mathbf{X}_{\text{Phase}}$. The magnitude spectra were converted to log-scale and normalized in energy. These spectrograms are treated as a two-dimensional (2D) array that exposes spatial patterns from which a machine can learn. Only the magnitude spectra were fed as inputs to the DNN model.

3.3 DNN architecture

The DNN with an autoencoder structure is trained to predict the target, which is the drum spectrogram. The layout of the CDAE using U-Net with dense block is shown in Fig. 2. The autoencoder is composed of two parts, *viz.* the encoder and decoder. The encoder computes the features on coarser time scales. Each hierarchical layer of the encoder has a series of convolution layers and a max-pooling layer. The filters used in the convolution layers extract the features from the incoming data. In each hierarchical layer in the encoder, the convolution layer retains the spectrogram size and increases

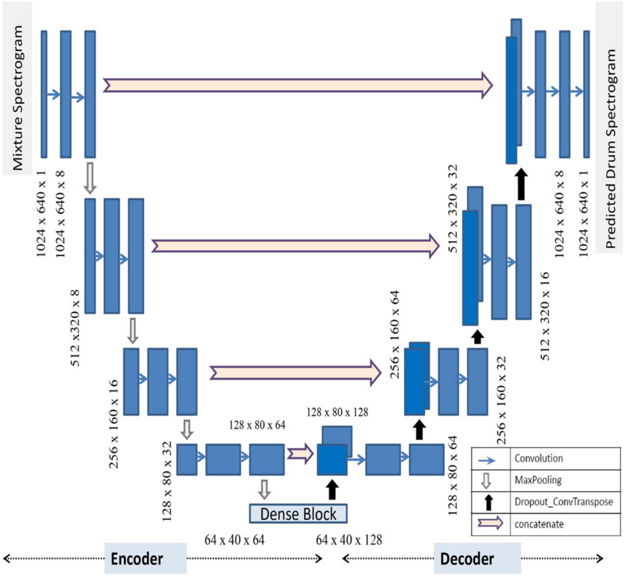

Fig. 2 Architecture of U-Net with dense block

[1] https://sigsep.github.io/datasets/dsd100.html.

[2] https://doi.org/10.5281/zenodo.1117372.

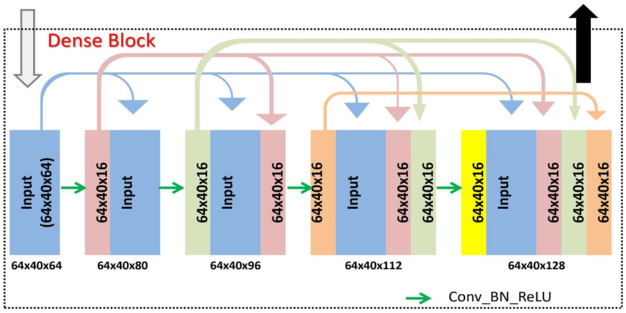

Fig. 3 Details of dense block

the number of channels. Each convolution layer consists of a kernel size of 3 × 3 and a padding factor of 1. Max-pooling is employed for the downsampling of the latent representation. The pooling layer has no learnable parameters, as it is used mainly for dimension reduction.

A dense block is introduced at the bottleneck of the autoencoder stage. The details of the dense block are shown in Fig. 3. It is composed of four blocks within a single block. Each block has 16 channels at the output. The features from preceding layers are concatenated with the successive layers and hence more information is conveyed from previous layers to subsequent layers. The input to the dense block has the size of 64 × 40 × 64. The input features are passed through convolution, batch normalization (BN) and rectified linear units (ReLU). The output features of the dense layer is fed to the decoder stage.

The decoder computes the local and high-resolution features. It possesses a series of transpose convolution followed by a convolution layer. The transpose convolution with a kernel size of 2 × 2 and a stride of 2 doubles the spectrogram size. The resultant array is then concatenated with the features from the encoder path. After concatenation, the convolution layer in the decoder retains the spectrogram size and decreases the number of channels. This process was repeated for each hierarchical layer. To prevent overfitting, a dropout of 0.5, is applied to each layer of the decoder. In the final layer, a 1 × 1 convolution is used to map the features to restore the original spectrogram size.

During downsampling in the encoder, many low-level details are lost as we force all the information to flow through the compression bottleneck. The direct skip connections used in U-Net between layers at the same hierarchical level allow information to flow directly from the encoder to the decoder layers.

Thus, the drum source features are extracted and the DNN model predicts the magnitude spectrogram $\widetilde{\mathbf{X}}_d$ of the drum signal. The DNN model is similarly trained with clean drum spectra replaced by mixture spectra and the DAE is tuned to predict $\widetilde{\mathbf{X}}$ of the mixture spectra.

3.4 Post-processing

The DNN is followed by the post-processing stage, in which a soft mask was calculated to estimate the magnitude spectrogram for the drum source. After decoding, corresponding denormalization and log-to-linear conversions were applied. Thus, we obtain the contribution of the drum signal in the mixture, which is a partial estimate of the magnitude of the drum spectrogram. The neural network does not have the constraint that the sum of the predicted masks is equal to the original mixture. To enforce the constraint, TF masking of the original mixture was performed. The soft mask $\mathbf{M}_d$ for drums was calculated using (1)

$$\mathbf{M}_d = \frac{\widetilde{\mathbf{X}}_d}{\widetilde{\mathbf{X}}} \tag{1}$$

The magnitude spectra $\widehat{\mathbf{X}}_d$ of the drum signal is

$$\widehat{\mathbf{X}}_d = \mathbf{M}_d \odot \mathbf{X}_{\text{Mag}} \tag{2}$$

computed using (2) where $\odot$ stands for element-wise multiplication. The estimated magnitude spectra are now combined with the phase spectra of the input mixture to retrieve the time-domain signals by applying ISTFT as given by (3).

$$\hat{x}_d(t) = \text{ISTFT}[\widehat{\mathbf{X}}_d \odot \mathbf{X}_{\text{Phase}}] \tag{3}$$

The separated drum signal in the time domain $\hat{x}_d(t)$ is estimated and compared with the ground truth of the original drum signal represented by $x_d(t)$.

4 Experiment

CDAE with U-Net architecture was trained with both the DSD and MUSDB datasets. The encoder input (2D array) of size 1024 × 640 is the magnitude spectrogram. As the 2D array is progressively processed through a series of convolution layers, the number of filters is increased to generate 8, 16, 32 and 64 feature maps consecutively. The spatial dimensions of these features are reduced to half the size by 2 × 2 max-pooling. At the bottle neck phase, the dense connectivity of the dense block extracts the essential features which are conveyed to the decoder. The final layer of the decoder computes the penult 3 × 3 convolution followed by 1 × 1 convolution to produce the predicted magnitude spectrogram of original array size.

During the training phase, clean drum spectra were used to train the network. 80% of the dataset is used for training while the rest was split into validation and test set. The binary cross-entropy loss function, which summarizes the average

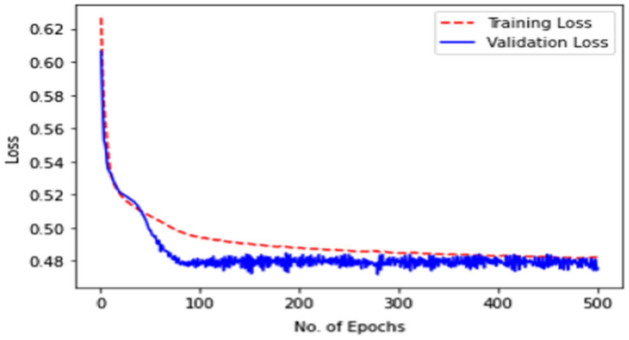

Fig. 4 Loss curve showing training and validation loss using MUSDB dataset

difference between the actual and predicted spectrogram was used in this model. The training was performed using the Adam optimizer [27] with a learning rate of 0.0001, for 500 epochs and with a batch size of 8. The hyperparameters of the Adam optimizer, such as $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1 \times 10^{-8}$ were chosen for training the network.

The training and validation loss curves for the U-Net with dense block using MUSDB dataset are presented in Fig. 4. The validation loss was used to evaluate the progress in training. During training, the weights of the kernel were initialized with "He-Normal" initializer [28]. The algorithm was implemented using TensorFlow—Keras. After training, the model was evaluated using the test dataset.

The performance of the proposed method was analyzed using the standard metric SDR [29], which provides a measure of distortion between the desired target and unwanted components and thus an overall assessment of the quality of the estimated sources. It is computed using *museval*[3] package generally employed in the music source separation evaluation campaign [30], in which the SDR is calculated using (4)

$$\text{SDR} = 10 \log_{10} \frac{\|s_{\text{target}}\|^2}{\|e_{\text{interference}} + e_{\text{noise}} + e_{\text{artifact}}\|^2} \tag{4}$$

The drum signal present in the input mixture is considered as s_{target}, the vocal and accompanying instrument tones as $e_{\text{interference}}$ and the background noise as e_{noise}. e_{artifact} is the forbidden distortion of sources or burbling artifacts.

Audio quality metrics are also based on perceptual perspective, which predict the difference between the reference signal and the distorted signal from the view point of human perception [31]. The PEASS (Perceptual Evaluation Methods for Audio Source Separation) toolkit[4] was used to measure the overall perceptual score (OPS) [32].

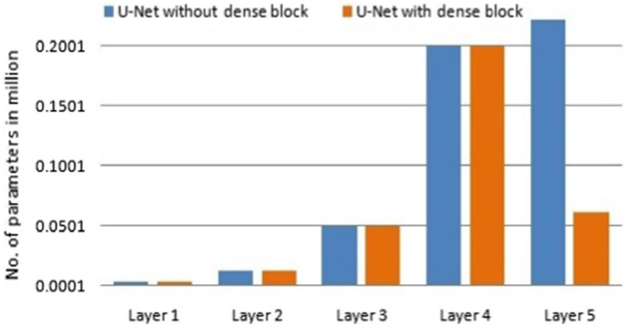

Fig. 5 Comparison of parameters in each layer of U-Net

Table 1 Comparison of average SDR (drums) and total parameters

Method	SDR in dB		Parameters
	DSD	MUSDB	
U-Net without dense block	5.36	5.41	485,673
U-Net with dense block	5.71	6.45	325,225

4.1 Ablation study

An ablation study was conducted to investigate the effectiveness of the dense block. The dense block was replaced by convolutional layers in the U-Net. The model was trained using the basic U-Net structure which has convolution filters at the bottle neck stage. The SDR values were computed and the number of parameters required were analyzed in both the cases. The number of parameters needed in each layer of U-Net is compared in the Fig. 5. The use of dense block has resulted in reduction of the parameters in layer 5. The U-Net without dense block needed 0.48 million parameters, while U-Net with the dense block needed 0.32 million parameters only. It is observed that the total number of parameters reduced significantly and the SDR values increased using the dense block, as listed in Table 1.

5 Results and discussions

A comparative study of the proposed method for drum signal separation was carried out with other state-of-the-art methods in terms of the average SDR and the number of parameters using both DSD and MUSDB databases. The results obtained are presented in Tables 2 and 3, respectively. The U-Net with dense block resulted in fewer parameters with at par SDR values. When the proposed method was tested with MUSDB, which had more training data, the SDR was found to increase.

The OPS computed for the DSD and the MUSDB database gives an average value of 38.3 and 42.5, respectively.

[3] https://github.com/sigsep/sigsep-mus-eval.

[4] https://gitlab.inria.fr/bass-db/peass.

Table 2 Performance comparison of average SDR (drums) on DSD dataset

Method	SDR (in dB)	Parameters (in million)
DeepConvSep [15]	2.40	5200.00
Blend [19]	4.00	8.71
MMDenseNet [20]	5.37	0.31
Proposed method using U-Net with dense block	5.71	0.32

Table 3 Performance comparison of average SDR (drums) on MUSDB dataset

Method	SDR (in dB)	Parameters (in million)
Wave-U-Net* [9,23]	4.16	10.20
Demucs* [11,23]	5.38	648.00
Meta-TasNet* [10,23]	5.91	45.50
DeepConvSep [15,23]	3.14	0.32
Spect U-Net [18,23]	4.66	9.84
Open-Unmix [23,24]	5.73	8.90
MMDenseNet [20]	6.27	0.33
MMDenseLSTM [21]	6.62	1.22
Sams-Net [23]	6.63	3.70
Proposed method using U-Net with dense block	6.45	0.32

*Denotes the waveform-based model

6 Conclusion

In this paper, a method for drum signal separation from a polyphonic music signal mixture using a denoising autoencoder with U-Net architecture and a dense block is presented. The autoencoder was trained and tested on the DSD and MUSDB datasets. The results show that while the drum source separation using the U-Net with a dense block is at par with other state-of-the-art methods in terms of SDR, the number of parameters for estimation is less in this architecture, making it computationally efficient. The ablation study has proved that the contribution of dense block improved the performance of the model. In the future, better separation can be achieved by training the network by employing dense blocks in other layers of the encoder and decoder.

References

1. Muller, M., Ellis, D.P., Klapuri, A., Richard, G.: Signal processing for music analysis. IEEE J. Sel. Top. Signal Process. **5**(6), 1088–1110 (2011)
2. Hyvarinen, A., Oja, E.: Independent component analysis: algorithms and applications. Neural Netw. **13**(4–5), 411–430 (2000)
3. Virtanen, T.: Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. IEEE Trans. Audio Speech Lang. Process. **15**(3), 1066–1074 (2007)
4. Gillet, O., Richard, G.: Transcription and separation of drum signals from polyphonic music. IEEE Trans. Audio Speech Lang. Process. **16**(3), 529–540 (2008)
5. Kotti, M., Ververidis, D., Evangelopoulos, G., Panagakis, I., Kotropoulos, C., Maragos, P., Pitas, I.: Audio-assisted movie dialogue detection. IEEE Trans. Circuits Syst. Video Technol. **18**(11), 1618–1627 (2008)
6. Helen, M., Virtanen, T.: Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine. In: 2005 13th European Signal Processing Conference, pp. 1–4. IEEE (2005)
7. Srinivasan, S., Roman, N., Wang, D.: Binary and ratio time-frequency masks for robust speech recognition. Speech Commun. **48**(11), 1486–1501 (2006)
8. Kadandale, V.S., Montesinos, J.F., Haro, G., Gómez, E.: Multi-channel U-NET for music source separation. In: 2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP), pp. 1–6. IEEE (2020)
9. Stoller, D., Ewert, S., Dixon, S.: Wave-U-Net: a multi-scale neural network for end-to-end audio source separation. arXiv preprint arXiv:1806.03185 (2018)
10. Samuel, D., Ganeshan, A., Naradowsky, J.: Meta-learning extractors for music source separation. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 816–820. IEEE (2020)
11. Défossez, A., Usunier, N., Bottou, L., Bach, F.: Music source separation in the waveform domain. arXiv preprint arXiv:1911.13254 (2019)
12. Huang, P.-S., Kim, M., Hasegawa-Johnson, M., Smaragdis, P.: Joint optimization of masks and deep recurrent neural networks for monaural source separation. IEEE/ACM Trans. Audio Speech Lang. Process. **23**(12), 2136–2147 (2015)
13. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
14. Uhlich, S., Giron, F., Mitsufuji, Y.: Deep neural network based instrument extraction from music. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2135–2139. IEEE (2015)
15. Chandna, P., Miron, M., Janer, J., Gómez, E.: Monoaural audio source separation using deep convolutional neural networks. In:

International Conference on Latent Variable Analysis and Signal Separation, pp. 258–266. Springer (2017)
16. Grais, E.M., Plumbley, M.D.: Single channel audio source separation using convolutional denoising autoencoders. In: 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp. 1265–1269. IEEE (2017)
17. Jansson, A., Humphrey, E., Montecchio, N., Bittner, R., Kumar, A., Weyde, T.: Singing voice separation with deep U-Net convolutional networks (2017)
18. Liu, J.-Y., Yang, Y.-H.: Denoising auto-encoder with recurrent skip connections and residual regression for music source separation. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 773–778. IEEE (2018)
19. Uhlich, S., Porcu, M., Giron, F., Enenkl, M., Kemp, T., Takahashi, N., Mitsufuji, Y.: Improving music source separation based on deep neural networks through data augmentation and network blending. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 261–265. IEEE (2017)
20. Takahashi, N., Mitsufuji, Y.: Multi-scale multi-band densenets for audio source separation. In: 2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), pp. 21–25. IEEE (2017)
21. Takahashi, N., Goswami, N., Mitsufuji, Y.: Mmdenselstm: an efficient combination of convolutional and recurrent neural networks for audio source separation. In: 2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC), pp. 106–110. IEEE (2018)
22. Satya, M.F., Suyanto, S.: Music source separation using generative adversarial network and U-Net. In: 2020 8th International Conference on Information and Communication Technology (ICoICT), pp. 1–6. IEEE (2020)
23. Li, T., Chen, J., Hou, H., Li, M.: Sams-Net: a sliced attention-based neural network for music source separation. In: 2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP), pp. 1–5. IEEE (2021)
24. Stöter, F.-R., Uhlich, S., Liutkus, A., Mitsufuji, Y.: Open-Unmix—a reference implementation for music source separation. J Open Source Softw **4**, 1667 (2019)
25. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-assisted Intervention, pp. 234–241. Springer (2015)
26. Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S.I., Bittner, R.: MUSDB18—a corpus for music separation (2017)
27. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
28. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034 (2015)
29. Vincent, E., Gribonval, R., Févotte, C.: Performance measurement in blind audio source separation. IEEE Trans. Audio Speech Lang. Process. **14**(4), 1462–1469 (2006)
30. Stöter, F.-R., Liutkus, A., Ito, N.: The 2018 signal separation evaluation campaign. In: International Conference on Latent Variable Analysis and Signal Separation, pp. 293–305. Springer (2018)
31. You, J., Reiter, U., Hannuksela, M.M., Gabbouj, M., Perkis, A.: Perceptual-based quality assessment for audio–visual services: a survey. Signal Process. Image Commun. **25**(7), 482–501 (2010)
32. Emiya, V., Vincent, E., Harlander, N., Hohmann, V.: Subjective and objective quality assessment of audio source separation. IEEE Trans. Audio Speech Lang. Process. **19**(7), 2046–2057 (2011)