International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A DenseU-Net framework for Music Source Separation using Spectrogram Domain Approach

Vinitha George E.*[1], V. P. Devassia[2]

Submitted:11/03/2024 **Revised:** 26/04/2024 **Accepted:** 03/05/2024

Abstract: Audio source separation has been intensively explored by the research community. Deep learning algorithms aid in creating a neural network model to isolate the different sources present in a music mixture. In this paper, we propose an algorithm to separate the constituent sources present in a music signal mixture using a DenseUNet framework. The conversion of an audio signal into a spectrogram, akin to an image, accentuates the valuable attributes concealed in the time domain signal. Hence, a spectrogram-based model is chosen for the extraction of the target signal. The model incorporates a dense block into the layers of the U-Net structure. The proposed system is trained to extract individual source spectrograms from the mixture spectrogram. An ablation study was performed by replacing the dense block with convolution filters to study the effectiveness of the dense block. The proposed method proves to be more efficient in comparison with other state-of-the-art methods. The experiment results to separate vocals, bass, drums and others show an average SDR of 6.59 dB on the MUSDB database.

Keywords: *Autoencoder, Convolutional Neural Network, Deep learning, DenseNet, Music source separation, ResNet, U-Net architecture*

1. Introduction

Audio signal processing is a research topic that concentrates on problems such as environmental sound classification [1], speech recognition [2], audio source separation, etc. Audio source separation involves automatically separating audio sources from a complex acoustic mixture. Complex musical signals are composed of vocals and instruments, which are usually polyphonic with 5-20 instruments. Musical instruments can be roughly classified into two groups, namely *harmonic* and *inharmonic* sounds. To recognize each sound, the features of music signals like pitch, duration and timbre are to be identified. Music heavily depends on its repetitions to cultivate an aesthetic feeling. Thus, effectively representing the repetitive patterns within the mixed signal could offer a better solution [3]. The application of music source separation includes instrument recognition, beat detection, polyphonic transcription from music signals, etc. [4].

The music separation methods can be classified into four variants:

1. Source separation based on the principle of computational auditory scene analysis (CASA) [5]: It aims to replicate the human auditory experience by emulating the principles of human ear perception, enabling the model to separate the target signal from mixed music signals.
2. Classical signal processing methods: They are principal component analysis (PCA) [6], independent component analysis (ICA) [7] and nonnegative matrix decomposition (NMF) [8]. NMF based approaches assume that mixture data is a linear combination of latent bases. It decomposes the music spectrum matrix into lower dimension matrices, which later help in identifying the sources for separation.
3. REPET algorithm: This models the repetition structure based on the spectrum in music [9].
4. Neural network based methods [10]: These methods rely on data-driven approaches employing supervised learning. Both the mixed signals and the individual instruments contributing to the music signal are accessible for training purposes. The design of deep learning algorithms tries to decrypt the human capability to identify audio sources.

Deep learning models for audio signal processing perform successive nonlinear mapping of the input mixture [10]. These models use 2D audio representations such as spectrograms and mel-frequency cepstrum coefficients (MFCC). In this study, a model based on spectrograms is favored. The merit of spectrogram representation is that the process of reconstruction is simpler compared to other feature representations, such as MFCC. To separate the sounds of various music components, features present in the spectrogram of the music mixture should be explored. Spectrogram-based models are trained using the time-frequency (TF) representation of the music mixture. Generally, spectrogram-based models have fewer trainable network parameters [11].

The main contributions of this paper are:

1. The development of a dense block based U-Net structure replacing the convolutional filters in the existing U-Net.

2. Comparison study of the revised structure with the existing method for the separation of constituent components from complex music signal.

The rest of this paper is structured as follows. A brief review of the spectrogram-based music signal separation techniques using neural networks is given in Section 2. The proposed music signal separation is unveiled in Section 3. The training and testing of the model are explained in Section 4. Section 5 showcases the findings and comparisons with other popular methods, offering insightful remarks. The summary and future directions are given in Section 6.

2. Related Work

The advent of the deep learning era has triggered interest in many signal processing techniques, like source separation. Several deep learning algorithms were developed to tackle audio source separation, especially speech and music [12-28]. The mapping between sources and mixture is a nonlinear relation. Hence, a deep learning network is a natural choice to address the problem of separation. Though few researchers performed the music source separation in the time-domain [12-14], spectrogram-domain is preferred because it provides more details. Deep learning models presented in the literature include fully connected neural network (FNN), convolutional neural network (CNN) [15], recurring neural network (RNN) [16] and combination of both CNN and RNN [17, 18].

Another variant of neural network used for music separation is the convolutional autoencoder [19]. It is a special kind of neural network used to reconstruct the input at the output layer. It has an encoder and a decoder. Yet another deep neural network (DNN) is U-Net, which is composed of such an encoder-decoder. Skip connections are incorporated between layers of the encoder and decoder at corresponding levels to transmit more detailed information from the encoder to the decoder. Such a structure was initially used for biomedical segmentation [29-31]. The advantage of the U-Net design is its high modularity and adaptability [28]. Jansson et al. [20] adopted U-Net for singing voice separation.

Dilated time-frequency denseNet was used for singing voice separation [22]. The denseNet expanded the receptive field more effectively by adding dilated convolution. Residual encoder decoder blocks are used in Deep ResU-Net [23] to improve music source separation by imparting 143 layers to U-Net. D3Net uses dilated convolutional blocks with dense connections. Densely connected convolutional blocks are used to allow the reuse of feature maps [25]. It allows maximum information flow while keeping the model size small [26].

Band-split RNN (BSRNN) was custom-tailored for processing high sample rate signals, offering precise partitioning and modeling of distinct frequency bands [27]. It incorporates prior knowledge about the source's characteristics to aid the selection of model hyperparameters. The music source separation employed in Generative Adversarial Networks (GAN) [28] utilized the U-Net model in the generator. The restoration of real-world audio signals corrupted by artifacts was performed using 1D GAN [32]; however, the authors suggest that spectral domain processing would play a crucial role in recovering the high-frequency components of the audio.

The drum source separation was performed in [33] to extract the drum signals from the mixture by introducing a single dense block at the bottleneck stage of the U-Net. Deploying the dense block in all the layers of the U-Net, culminated in a DenseU-Net. Vocal separation from the music mixture was performed using this DenseU-Net [34]. In this paper, all the constituent components of the complex music signal *viz.* the vocals, drums, bass and other instruments are separated.

3. Proposed System

We propose in this section, music source separation using DenseU-Net. Our goal is to develop a deep learning algorithm to extract multiple audio files, such as vocals, bass, drums and the rest of other instruments. The usual method involves constructing dedicated models for individual source separations. The suggested framework has four phases: pre-processing, training/testing, post processing, and performance assessment. The block diagram of the process of music signal separation from a polyphonic music mixture using a DNN is illustrated in Fig. 1.

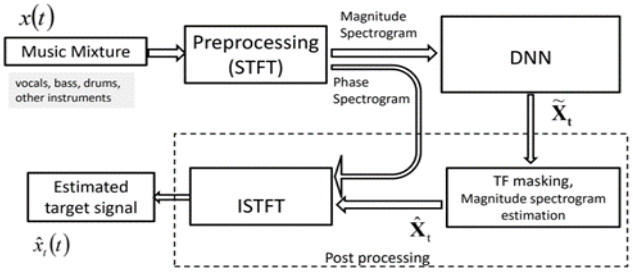

Fig.1. Block diagram of music source separation

The complex music signal intended for separation is derived from a standard audio database. In the preprocessing phase, the time-domain signal $x(t)$ undergoes conversion into the spectrogram domain. The DNN takes the magnitude spectrogram as its input, while preserving the phase spectra for subsequent synthesis. Employing the U-Net architecture, the DNN predicts the target spectrogram. This predicted spectrogram undergoes post-processing at a later stage to reconstruct the time-domain source signal, denoted as $\hat{\boldsymbol{x}}_{\boldsymbol{t}}(t)$.

The DNN model facilitates the prediction of the individual spectrogram from the mixture spectrogram. Soft TF masking is utilized to enhance the quality of the predicted source. Once the TF mask is computed, it is integrated with the magnitude spectrogram of the mixture to estimate the source spectra. The phase spectrogram of the mixture is merged with the estimated magnitude spectrogram to restore the estimated source waveform using inverse STFT (ISTFT). In the following paragraph, a detailed description of the denseU-Net and the dataset used for the experiment are explained. The experimental procedure of audio preprocessing and the postprocessing performed later for audio restoration are also explained.

3.1. Dataset

The MUSDB dataset was used in this study. The MUSDB database is a professionally recorded music source, available in stereo format, with a sample rate of 44.1 kHz [35]. It contains 150 professionally recorded songs, of which 100 are used as the training set and the rest as the test set. Each song consists of a mixture $x(t)$ and its four sources, *viz.* the vocals$(x_v(t))$, the drum signal $(x_d(t))$, the bass $(x_b(t))$ and other instruments. The task is to separate the vocals , drums, bass and the rest of the other instruments from the mixture.

3.2. Data Preprocessing

The preparation of the dataset for training the DNN is done at the preprocessing stage. The stereo wave songs were converted to mono by averaging both channels. The resultant audio signal was converted to the corresponding spectrogram using the STFT. A Hanning window of length 2048 samples was chosen with a hop size of one-fourth the window length.

In a spectrogram, time and frequency are represented on the horizontal and vertical axis respectively. Brightness in the spectrogram image attributes to the strength of a frequency component at each time frame. The magnitude spectrogram contains most of the information of the audio signal. Hence, only the magnitude spectra were fed as inputs to the DNN model.

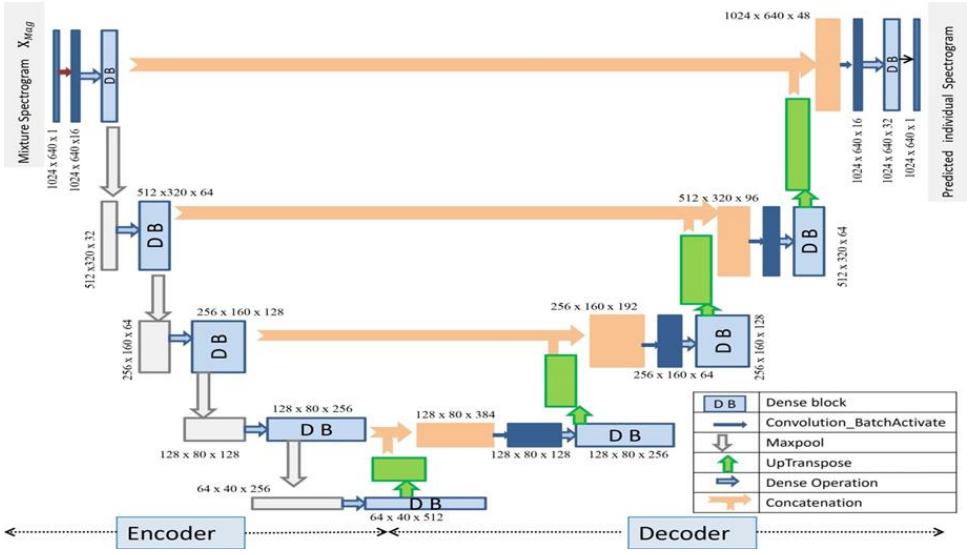

Fig.2. Architecture of DenseU-Net

3.3. DNN Architecture

The DNN with DenseU-Net is trained to predict the target spectrogram. It learns suitable spectral-temporal features of the vocal/drum/bass/other instruments that are hidden in the 2D array of the mixture spectrogram. The architecture of the DenseU-Net model used in the DNN model is shown in Fig. 2. The DenseU-Net comprises two components: namely, the encoder and the decoder.

3.3.1 The Encoder

This part receives the mixture spectrogram and the ground truth source spectrogram as input during the training phase. The target features present in the TF frames of the mixture spectrogram are extracted. The coarser details in the frames are captured by the encoder. It comprises of a dense block and a max-pooling layer at each step. Convolution layers with a 3×3 kernel size and a padding factor of 1 are utilized in each hierarchical layer. Thus, in every layer, the

spectrogram size is retained while the number of filters is increased. The rectified linear units (ReLU) activation function is applied due to the sparsity of magnitude spectrograms in audio signals. This is followed by batch normalization for network stability post-convolution [36]. Max-pooling ensures down-scaling to find the latent representation. Table 1 provides details on the filter configuration in the encoder. Conv2D (3×3) signifies 2D convolution with a 3×3 kernel size. The encoder progressively increases the number of filters to explore the depth of the feature space, facilitating the learning of diverse levels of global abstract structures.

Table 1. Configuration of encoder using dense block

Process	Configuration	Output Size
Conv2D(3×3)	No. of filters = 16	1024 x 640 x 16
BatchNormalization_Activation		
Dense Block		1024 x 640 x 32
Max-Pooling2D(2,2)	k = 16/4	512 x 320 x 32
Dense Block	k = 32/4	512 x 320 x 64
Max-Pooling2D(2,2)		256 x 160 x 64
Dense Block	k = 64/4	256 x 160 x 128
Max-Pooling2D(2,2)		128 x 80 x 128
Dense Block	k = 128/4	128 x 80 x 256
Max-Pooling2D(2,2)		64 x 40 x 256
Dense Block	k = 256/4	64 x 40 x 512

3.3.2 Dense Block

To enable the network to grasp a broader range of features, a sequence of convolutional layers is employed. However, this approach often leads to the learning of redundant features. To address this issue and optimize information flow within the network, densely connected convolutions are utilized. This strategy ensures the acquisition of a diverse set of features by leveraging the collective knowledge gained from previous layers, thus preventing redundancy. The configuration of the dense block, as illustrated in Fig. 3, comprises four blocks nested within a single block. At each layer, features from preceding blocks are concatenated with the subsequent block, facilitating the transfer of more information from the previous block to the succeeding one. The growth rate, denoted as '*k'*, determines the number of features added to the subsequent block in each layer. For instance, in layer 3, with a growth rate of 16, 16 feature maps are introduced to the subsequent block. The input features undergo convolution, batch normalization, and ReLU activation functions, as illustrated in Fig. 3. The feature maps from each layer in the encoder are linked with their corresponding counterparts in the decoder through concatenation.

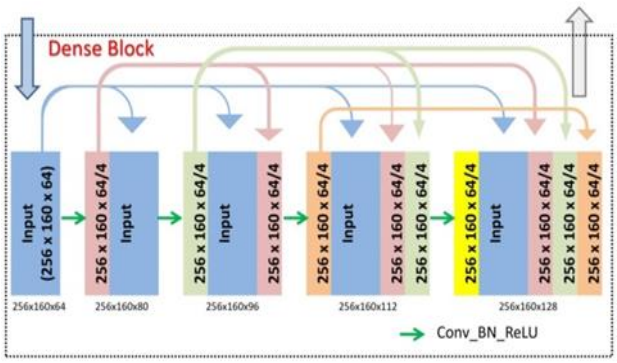

Fig. 3. Details of a Dense Block (layer 3)

3.3.3. The Decoder

The decoder is responsible for computing local and high-resolution features. In each step of the decoding path, a transpose convolution function is applied to the output of the preceding layer. Utilizing a 2 x 2 stride in the transpose convolution doubles the spectrogram size of the resulting array. Subsequently, this array is concatenated with features from the encoder path. This is followed by a dense block and a convolution layer. The incorporation of batch normalization expedites the convergence speed of the network. Post-concatenation, the convolution layer in the decoder maintains the spectrogram size through a 1 x 1 kernel size. However, the number of channels is reduced. The features are then passed through a dense block to extract crucial information. This process is replicated for each hierarchical layer. In the final layer, a 1 x 1 convolution is utilized to map the features and restore the original size of the spectrogram. The details of the filter configuration used in the decoder are listed in Table 2.

Table 2. Configuration of decoder using dense block

Process	Configuration	Output Size
Conv2Dtranspose(3×3)	No. of filters = 128	128 x 80 x 128
Concatenate		128 x 80 x 384
Conv2D(1×1)	No. of filters = 128	128 x 80 x 128
BatchNormalization_Activation		128 x 80 x 128
Dense Block	k = 128/4	128 x 80 x 256
Conv2Dtranspose(3×3)	No. of filters = 64	256 x 160 x 64
Concatenate		256 x 160 x 192
Conv2D(1×1)	No. of filters = 64	256 x 160 x 64
BatchNormalization_Activation		256 x 160 x 128
Dense Block	k = 64/4	256 x 160 x 128
Conv2Dtranspose(3×3)	No. of filters = 32	512 x 320 x 32
Concatenate		512 x 320 x 96
Conv2D(1×1)	No. of filters = 32	512 x 320 x 32
BatchNormalization_Activation		512 x 320 x 64
Dense Block	k = 32/4	512 x 320 x 64
Conv2Dtranspose(3×3)	No. of filters = 16	1024 x 640 x 16
Concatenate		1024 x 640 x 48
Conv2D(1×1)	No. of filters = 16	1024 x 640 x 16
BatchNormalization_Activation		1024 x 640 x 32
Dense Block	k = 16/4	1024 x 640 x 32
Conv2D(1×1)	No. of filters = 1	1024 x 640 x 1

The U-Net's unique skip connections bridge information gaps between encoder and decoder layers. Skip connections ensures that the encoder features influence the learning process throughout the network. This allows the network to capture the essential features desired for extracting the target features from the music mixture. Skip connections also alleviate the vanishing gradient problem. Thus, by training with clean ground truth signal spectrograms, the DNN model learns to predict their magnitude spectrogram $\widetilde{\mathbf{X}_t}$. Later, the training process involves replacing clean target spectra with mixture spectrogram. This prompts the model to predict mixture spectrogram $\widetilde{\mathbf{X}}$ needed during post-processing.

3.3.4 Postprocessing

The DNN predicts the spectrogram of the target. The postprocessing stage follows the DNN framework. This stage ensures that the time-domain signal is retrieved from spectrogram. During post-processing, a soft mask is calculated to estimate the magnitude spectrogram for the target source. The soft mask enables us to determine the contribution of each target signal in the mixture. Thus a partial estimate of the magnitude of the target spectrogram is obtained. However, the sum of the estimated spectra may not match the original mixture signal, as the DNN does not ensure that the sum of the predicted masks is equal to the original mixture. To overcome the constraint, TF masking was performed. The soft mask $\boldsymbol{M_t}$ for each target is calculated using (1)

$$\boldsymbol{M_t} = \frac{\widetilde{\mathbf{X}_t}}{\widetilde{\mathbf{X}}} \qquad (1)$$

The magnitude spectra $\widehat{\mathbf{X}}_\mathbf{t}$ of each target signal is computed using (2)

$$\widehat{\mathbf{X}_\mathbf{t}} = \boldsymbol{M_t} \odot \mathbf{X_{Mag}} \qquad (2)$$

where $\odot$ stands for element-wise multiplication. In order to retrieve the time-domain signal of the target, the magnitude spectra obtained from (2) is combined with the phase spectra of the mixture spectrogram. The inverse STFT is applied to the resultant 2D array as given by (3).

$$\widehat{\boldsymbol{x}}_\boldsymbol{t}(t) = \mathbf{ISTFT}\left[\widehat{\mathbf{X}}_\mathbf{t} \odot \mathbf{X}_{\boldsymbol{Phase}}\right] \qquad (3)$$

The separated target $x_t(t)$ in the time domain are estimated and compared with the ground truth of the original signals.

4. Training and Testing the DenseU-Net model

The DenseU-Net model was trained with clean target spectra to identify the target features present in the music mixture. The binary cross entropy loss is used to determine the loss during the training phase. This loss function provides the average difference between the ground truth spectrogram and the predicted spectrogram. The optimizer chosen for the training was the Adam optimizer [37] with a learning rate of 1×10^{-4}, executed for 300 epochs and a batch size of 4. The hyperparameters of the Adam optimizer, such as $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\in = 1 \times 10^{-8}$ were chosen for training the network. 20% of the training set was chosen to form the validation set.

The progress of training was evaluated using the validation loss. During training, the weights of the kernel were initialized with a 'He-Normal' initializer [38]. After the training phase, the model was tested with the test dataset. During the testing process, only the mixture signal from the test dataset is fed to the separation system. The test set was preprocessed using the same method as the training dataset.

The separation performance was analysed using a standard metric employed in the music source separation evaluation campaign [39]. It is the SDR, 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 [40]. The performance index is evaluated in terms of the SDR, computed using (4).

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

While the vocal signal is the desired target, s_{target}, the drum, bass and accompanying instrument tones were considered as $e_{interference}$. Similarly, while drum is targeted, other sources like bass, vocals etc. contribute to $e_{interference}$. The background noise is e_{noise}. $e_{artifact}$ is the forbidden distortion of sources or burbling artifacts.

4.1. Ablation study

An ablation study was conducted to assess the contribution of the dense block. This involves replacing the dense block with a convolutional layer in the U-Net. Thus, a baseline model is the original U-Net structure, which uses convolutional filters throughout. The average SDR was found to be 5.16 dB. When the convolution filter was replaced by a dense block at the bottleneck of U-Net, there is a significant rise in SDR. The dense block improves the flow of information and gradients throughout the network. So it is easier to train the U-Net with dense blocks. Hence the convolutional filters in all layers were replaced by dense blocks. The inclusion of dense blocks in all layers enhanced the performance of the network resulting in a better prediction of the target mask. The SDR is found to be 6.59 dB. The denseU-Net shows a mettle by steady increase in SDR, as shown in Fig. 4.

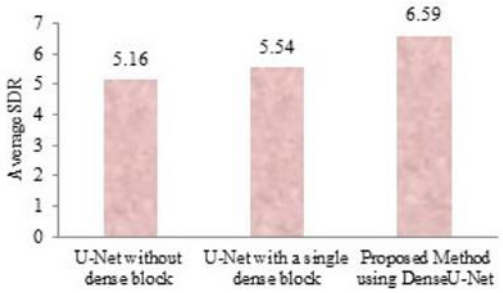

Fig. 4. Performance comparison of U-Net

5. Results and Discussions

The performance of the proposed method for music signal separation was compared with other state-of-the-art methods in terms of the average SDR. A comparative study using the MUSDB dataset is presented in Table 3. An ablation study was conducted by removing the dense blocks in U-Net and replacing them with convolution filters. The SDR was found to be 5.16 dB. When a single dense block was introduced at the bottleneck stage of the U-Net, the SDR improved by 0.38 dB. When the DenseU-Net model, i.e., U-Net with the dense block in all the layers is employed, the average SDR improved to 6.59 dB. The SDR measures of individual sources are presented in Fig. 5. Thus, with a dense block, there is a better identification and extraction of target features immersed in the music mixture.

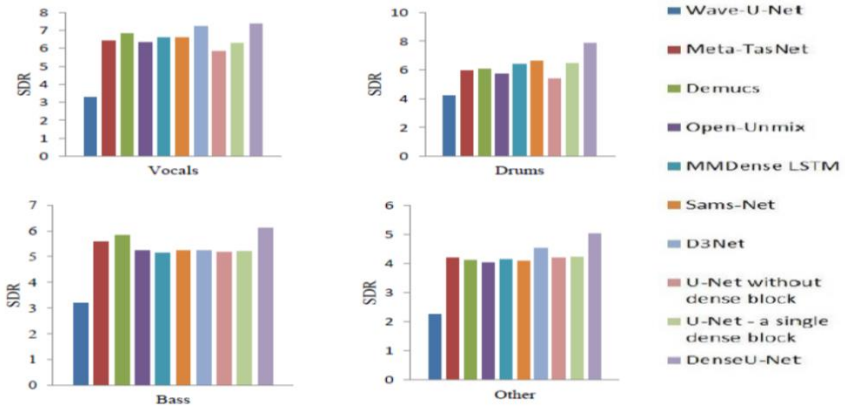

Fig. 5. Comparison of SDR with state-of-the-art methods

Table 3. Performance comparison in terms of average SDR on MUSDB dataset

Method	Average
Wave-U-Net* [12]	3.23
Meta-TasNet *[13]	5.52
Demucs *[14]	5.58
Open-Unmix [24]	5.36
MMDense LSTM [17]	5.58
Sams-Net [28]	5.65
D3Net [25]	6.01
U-Net without dense block [34]	5.16
U-Net with a single dense block [34]	5.54
Proposed Method using DenseU-Net	6.59

* denotes the waveform-based model

6. Conclusions

This study has introduced a robust method for music source separation through the implementation of DenseU-Net architecture. The utilization of this model, as demonstrated through comprehensive training and testing on the MUSDB datasets, has exhibited promising outcomes in complex music source separation, particularly in terms of SDR. However, the potential for further enhancement and fine-tuning exists. For instance, future research could explore the impact of adjusting the growth rate in each layer, providing an avenue for optimizing the

model's performance and potentially uncovering configurations that better suit a specific case. The attention mechanism can be introduced in the skip connection to improve feature propagation from encoder to decoder. Exploring and implementing these suggested improvements would undoubtedly contribute to the ongoing evolution of DenseU-Net for music source separation, paving the way for advancements in the field.

References

[1] G. Ozmen, I. A. Ozkan, I. Seref, S. Tasdemir, C¸ Mustafa and E. Arslan, "Sound analysis to recognize cattle vocalization in a semi-open barn," Gazi Muhendislik Bilimleri Dergisi " , vol. 8, no. 1, pp. 158–167, 2022.

[2] Binjaku, K., Janku, J. and Meçe, E.K., "Identifying Low-Resource Languages in Speech Recordings through Deep Learning," In 2022 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), pp. 1-6, IEEE, September, 2022.

[3] Yuan, W., Wang, S., Li, X., Unoki, M. and Wang, W., 2019. A skip attention mechanism for monaural singing voice separation. IEEE Signal Processing Letters, 26(10), pp.1481-1485.

[4] Klapuri, T. Virtanen, and T. Heittola, "Sound source separation in monaural music signals using excitation-filter model and EM algorithm," in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '10), pp. 5510–5513, March 2010.

[5] Zhao, Y., Wang, D., Johnson, E.M. and Healy, E.W., "A deep learning based segregation algorithm to increase speech intelligibility for hearing-impaired listeners in reverberant-noisy conditions," The Journal of the Acoustical Society of America, 144(3), pp.1627-1637, 2018.

[6] Huang, Po-Sen, et al. "Singing-voice separation from monaural recordings using robust principal component analysis." 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2012.

[7] Hyvärinen A, Oja E, "Independent component analysis: algorithms and applications," Neural networks, vol. 13(4-5), pp. 411- 430, June 2000.

[8] T. Virtanen, "Monaural sound source separation by non-negative matrix factorization with temporal continuity and sparseness criteria," IEEE Transactions on Audio, Speech and Language Processing, vol. 15, no. 3, pp. 1066–1074, 2007.

[9] Doğan, S.M. and Salor, Ö.,"Music/singing voice separation based on repeating pattern extraction technique and robust principal component analysis," In 2018 5th International Conference on Electrical and Electronic Engineering (ICEEE) (pp. 482-487), IEEE, May 2018.

[10] P. S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," IEEE/ACM Trans. ASLP, vol. 23, no. 12, pp. 2136-2147, 2015.

[11] Kadandale, Venkatesh S., Juan F. Montesinos, Gloria Haro, and Emilia Gomez, "Multi-channel U-Net for Music Source Separation," in IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP), pp. 1-6, IEEE, 2020.

[12] D. Stoller, S. Ewert, and S. Dixon, "Wave-u-net: A multiscale neural network for end-to-end audio source separation," in Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, pp. 334 – 340, 2018.

[13] D. Samuel, A. Ganeshan, and J. Naradowsky, "Meta-learning Extractors for Music Source Separation," in Proceedings of the 45th International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020), May 2020.

[14] Défossez A, Usunier N, Bottou L, Bach F, "Music source separation in the waveform domain," arXiv preprint arXiv:1911.13254, Nov 27, 2019.

[15] Chandna P, Miron M, Janer J, Gomez 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] S. Uhlich, M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahashi, and Y. Mitsufuji, "Improving music source separation based on deep networks through data augmentation and network blending," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 261–265, March 2017.

[17] N. Takahashi, N. Goswami, and Y. Mitsufuji, "MMDenseLSTM: An efficient combination of convolutional and recurrent neural networks for audio source separation," in 16th International Workshop on Acoustic Signal Enhancement (IWAENC), pp. 106–110, IEEE, 2018.

[18] J.-Y. Liu and Y.-H. Yang, "Dilated convolution with dilated GRU for music source separation," in International Joint Conferences on Artificial Intelligence Organization, (IJCAI), 2019.

[19] E. M. Grais and M. D. Plumbley, "Single channel audio source separation using convolutional denoising autoencoders," in IEEE global conference on signal and information processing (GlobalSIP), pp.1265 -1269, Nov. 2017.

[20] Jansson A., Humphrey E.J., Montecchio N., Bittner R., Kumar A., and Weyde T., "Singing voice separation with deep U-Net convolutional networks,"

in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), pp. 323 – 332, 2017.

[21] Giorgio Fabbro, Stefan Uhlich, Chieh-Hsin Lai, Woosung Choi, Marco Martinez-Ramirez, Weihsiang Liao, Igor Gadelha, Geraldo Ramos, Eddie Hsu, Hugo Rodrigues, et al., "The sound demixing challenge 2023—Music demixing track," arXiv preprint arXiv:2308.06979, 2023.

[22] W.-H. Heo, H. Kim, and O.-W. Kwon, "Source separation using dilated time-frequency DenseNet for music identification in broadcast contents," Applied Sciences, vol. 10, no. 5, pp. 1727, Mar. 2020.

[23] Kong, Q., Cao, Y., Liu, H., Choi, K. and Wang, Y., "Decoupling magnitude and phase estimation with deep resunet for music source separation," in 22nd International Society for Music Information Retrieval Conference, 2021.

[24] Fabian-Robert Stöter, Stefan Uhlich, Antoine Liutkus, and Yuki Mitsufuji, "Open-unmix – a reference implementation for music source separation," Journal of Open Source Software, 4(41), pp.1667, 2019.

[25] N. Takahashi and Y. Mitsufuji, "D3Net: Densely connected multidilated DenseNet for music source separation," arXiv preprint arXiv:2010.01733, 2020.

[26] G. Huang, Z. Liu, K. Q. Weinberger, "Densely connected convolutional networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4700–4708, 2016.

[27] Luo, Y. and Yu, J., 2022. Music source separation with band-split rnn. arXiv preprint arXiv:2209.15174.

[28] Li Tingle, Jiawei Chen, Haowen Hou, and Ming Li., "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.

[29] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in International Conference on Medical Image Computing and Computer-Assisted Intervention," pp. 234–241, Springer, 2015.

[30] Yue Cao, Shigang Liu, Yali Peng, and Jun Li., "Denseunet: densely connected unet for electron microscopy image segmentation," IET Image Processing, 14(12):2682–2689, 2020.

[31] Jayashree, P., Rajesh, P., Amol, D., Nihar, R., Mubin, T, "Gradient bald vulture optimization enabled multi-objective Unet++ with DCNN for prostate cancer segmentation and detection. Biomed. Signal Process. Control," 87, 105474 2024. https://doi.org/10.1016/j.bspc.2023.105474.

[32] Ince T, Kiranyaz S, Devecioglu O C, Khan M S, Chowdhury M, Gabbouj M, "Blind Restoration of Real-World Audio by 1D Operational GANs," arXiv preprint arXiv:2212.14618. 2022.

[33] E. V. George and V. P. Devassia, "A novel U-Net with dense block for drum signal separation from polyphonic music signal mixture," Signal, Image Video Process., vol. 17, no. 3, pp. 627–633, Apr. 2023.

[34] Vinitha George E and V P Devassia, "A DenseU-Net for separation of vocals from polyphonic music signal mixture", Grenze International Journal of Engineering and Technology, vol.9, no. 1, pp. 2648-2655, Jan 2023

[35] Zafar Rafii, Antoine Liutkus, Fabian-Robert Stoter, Stylianos Ioannis Mimilakis, and Rachel Bittner, "The MUSDB18 corpus for music separation," Dec. 2017.

[36] S. Ioffe and C. Szegedy., "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in International conference on machine learning, PNLR, pp. 448–456, 2015.

[37] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in Proceedings of the IEEE international conference on computer vision ICCV, pp. 1026-1034, 2015.

[39] Stoter, Fabian-Robert, Antoine Liutkus, and Nobutaka Ito, "The 2018 signal separation evaluation campaign," in International Conference on Latent Variable Analysis and Signal Separation, pp. 293 - 305, Springer, 2018.

[40] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," IEEE Transactions on Audio, Speech and Language Processing, vol. 14, no. 4, pp. 1462–1469, July 2006.