applied sciences

MDPI

Article

# Source Separation Using Dilated Time-Frequency DenseNet for Music Identification in Broadcast Contents

**Woon-Haeng Heo [1], Hyemi Kim [2] and Oh-Wook Kwon [1,*]**

[1] School of Electronics Engineering, Chungbuk National University, Chungdae-ro 1, Seowon-Gu, Cheongju, Chungbuk 28644, Korea; whheo89@cbnu.ac.kr

[2] Creative Content Research Division, Electronics and Telecommunications Research Institute, 218 Gajeong-ro, Yuseong-gu, Daejeon 34129, Korea; miya0404@etri.re.kr

* Correspondence: owkwon@cbnu.ac.kr; Tel.: +82-43-261-3374

Received: 19 December 2019; Accepted: 24 February 2020; Published: 3 March 2020


**Abstract:** We propose a source separation architecture using dilated time-frequency DenseNet for background music identification of broadcast content. We apply source separation techniques to the mixed signals of music and speech. For the source separation purpose, we propose a new architecture to add a time-frequency dilated convolution to the conventional DenseNet in order to effectively increase the receptive field in the source separation scheme. In addition, we apply different convolutions to each frequency band of the spectrogram in order to reflect the different frequency characteristics of the low- and high-frequency bands. To verify the performance of the proposed architecture, we perform singing-voice separation and music-identification experiments. As a result, we confirm that the proposed architecture produces the best performance in both experiments because it uses the dilated convolution to reflect wide contextual information.

**Keywords:** source separation; DenseNet; broadcast contents; music identification; dilated convolution

## 1. Introduction

Background music is a sensitive issue concerning copyright. This issue occurs in a variety of places such as broadcasting, music stores, online music streaming services and so on. To charge for copyright, it is important to know what title the background music is and how long it has been played. It is inaccurate and takes a lot of time and work to manually record the title and the playing time of background music. In order to solve this problem, automatic music identification and music section detection techniques are required. In this study, we perform automatic music identification on broadcast content. Due to the nature of broadcast content, background music is mostly mixed with speech louder than music. This characteristic results in lowering automatic music-identification performance. Therefore, we attempt to apply background music separation technique before automatic music identification.

Music signal separation was conventionally done by the traditional methods used in blind source separation (BSS) [1] such as independent component analysis (ICA) [2], non-negative matrix factorization (NMF) [3], and sparse component analysis (SCA) [4]. For monophonic music source separation, which is the same task as in this paper, NMF showed better performance than ICA and SCA [1]. However, NMF does not yield good separation performance in real environmental conditions because the NMF algorithm has inherently linear characteristics [5]. Recently, deep learning-based music source separation algorithms achieved good performance and outperformed the NMF algorithm [6–9]. In addition, deep learning-based music source separation algorithms have the advantage that they do not have the permutation problem [10] because they are trained

to directly produce the target signal. Grais et al. [6] used the feed-forward neural network (FNN) architecture in speech source separation task and achieved better performance than NMF. This means that deep learning is suitable for replacing the conventional algorithms. Nugraha et al. [7] and Uhlich et al. [8] experimented with music source separation using the FNN architecture, which is the basic structure in deep learning. They used different datasets, but they showed better performance compared to NMF. After then, a lot of new neural network architectures were applied to audio source separation: convolutional neural network (CNN), recurrent neural network (RNN), bidirectional long-term memory (BLSTM), and so on [11,12].

In the time-frequency spectrogram domain, the time-axis frames are used as input to the RNN or BLSTM architecture [11,12]. In a previous study [12], music source separation was performed with a BLSTM architecture. As the layer of BLSTM was stacked up to 3 layers, it showed good performance. When the number of layers was equal to FNN, BLSTM showed better performance than FNN. Data augmentation and Wiener filter for eliminating the stationary noise of the estimated target spectrogram further improved performance. Also, it showed better performance in a multi-channel environment than single-channel. The best performance in this experiment was obtained by weighted summation of estimated signals from BLSTM and FNN.

Recently, the CNN-based U-Net [13], stacked hourglass [14], and dense convolutional network (DenseNet) [15], which showed good performance in the image domain [16–18], were also successfully applied to audio source separation. The CNN-based U-Net [13] and stacked hourglass network [14] have an encoder-decoder style structure. Bottleneck features are generated between encoder-decoder processes. In U-Net, the feature map of each encoder is concatenated with the decoder feature map of the same resolution. The stacked hourglass network adds the feature map of the encoder to the decoder feature map of the same resolution via convolution. The connections between these encoders and decoders are advantageous for the transmission of information and the transmission of gradients. A stacked hourglass network consists of small network modules with an encoder-decoder structure to form the whole structure. The desired signal can be estimated between each module, and the loss can be calculated at each module output for back-propagation. This intermediate supervision improves the learning speed and network performance.

In the previous methods, the magnitude of the target was estimated in the spectrogram domain and the output signals were reconstructed by using the phase of the mixture [6–9,11–15]. However, the reconstructed signal in the time domain is distorted. To compensate for this distortion, Stoller et al. [19] used an end-to-end Wave-U-Net model in the time domain by modifying the U-Net architecture. Their structure is different from the U-Net with a similar shape in that it changes from the time-frequency domain to the time domain and uses 1-dimensional (1d) convolution instead of two-dimensional (2d) convolution. The end-to-end Wave-U-Net model in the time domain usually has a deeper structure to achieve good performance and has the disadvantage that it is difficult to converge. Although Wave-U-Net showed better performance than U-Net, it has lower performance than DenseNet-based architecture and BLSTM structure in the time-frequency domain [18].

The DenseNet-based architecture has recently been shown to perform well in audio source separation tasks [15,20,21]. Multi-scale multi-band DenseNet (MMDenseNet) [15] and MMDenseNet combined with LSTM (MMDenseLSTM) [20] are music source separation studies using DenseNet. Both architectures are based on CNN and are of an encoder-decoder style like the U-Net and stacked hourglass. The architecture of MMDenseNet was designed to parallel MDenseNet for the low-, high-, and whole-frequency band of the spectrogram, respectively. The architecture of MMDenseLSTM closely resembles the structure of MMDenseNet and places the LSTM behind the dense block. Because DenseNet is based on CNN, it learns pattern for image input, and LSTM learns pattern for time-series data, it has the advantage of learning different patterns and supplementing each other. MMDenseLSTM shows better performance than MMDenseNet, which shows the state-of-the-art performance at audio source separation. Since they are all based on the CNN architecture structured in an encoder-decoder style through down-sampling and up-sampling, bottleneck features are obtained through encoding.

This is the reason why designing this structure expands the receptive field, especially in the spectrogram domain. In neuroscience, the receptive field is the local area of the previous layer output where neurons are connected. Neurons in the visual cortex exhibit local features in the early visual layer and more complex patterns in the deep layer [22]. This is an important element of CNN that has inspired CNN [23].

In our study, we propose a dilated time-frequency DenseNet architecture to expand the receptive field effectively. We add a time-dilated convolution [24] which is a frame dilation rate of 2 and a frequency-dilated convolution which is a frequency dilation rate of 2 to the DenseNet. The previous CNN-based architectures expanded their receptive field with an encoder-decoder style architecture, but we expanded the receptive field more effectively by adding dilated convolution. The time- and frequency-dilated convolution systematically aggregate multi-scale contextual information of the time and frequency axes of the spectrogram. MMDenseNet is designed to place the MDenseNet model structure in parallel on each band of the spectrogram divided in half. In MMDenseNet structure, information exchange between models of each band is performed only in the last few layers, which makes it difficult to share information between each band, resulting in distortion in the output. Therefore, the proposed architecture applies a different convolution for each frequency band. The proposed architecture is shown to have the best performance in both separation and identification tasks compared with the previous architectures: U-Net, Wave-U-Net, MDenseNet, and MMDenseNet.

In the previous work [25] done by ourselves, we studied music detection using convolutional neural networks with a Mel-scale kernel from broadcast contents. The difference between our previous work and this work is in the type of task. Whereas the previous work is a classification task for music detection, this work is a regression task to estimate the music signal itself. To combine the previous work and this work for the purpose of music identification, the integrated system should be configured in order of music source separation, music detection, and music identification. In addition, in order to construct the integrated system, it is necessary to jointly optimize the deep learning architecture of source separation and music detection using the same training data. This issue will be studied later in another work.

In Section 2, DenseNet and the baseline architecture, which applies the DenseNet to audio source separation, are introduced. In Section 3, the overall proposed architecture is described. Two experiments and results are presented in Section 4. The first experiment is singing voice separation on the open resource, and the second one is music identification after source separation from our dataset. Finally, the overview and conclusion are described in Section 5.

## 2. Baseline

The baseline of our study is MDenseNet. We describe the DenseNet used for MDenseNet and why it is better than the previous CNN architectures. Next, we will briefly describe the baseline architecture based on DenseNet.

### 2.1. DenseNet

DenseNet [18] is a dense block structure consisting of composite functions as shown in Figure 1. A composite function consists of three consecutive operations: batch normalization (BN) [26], followed by a rectified linear unit (ReLU) [27], and a $3 \times 3$ convolution (Conv), as shown in Figure 2.

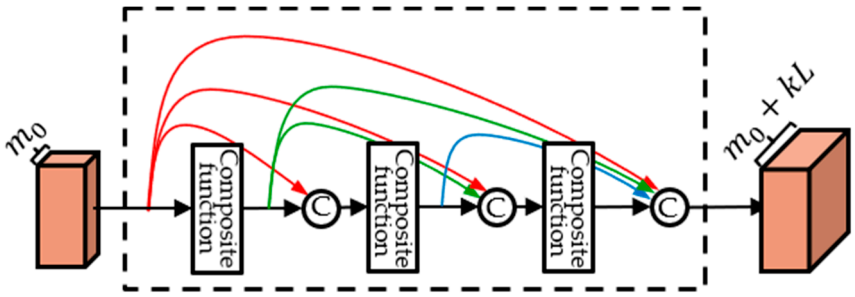


**Figure 1.** Dense block.

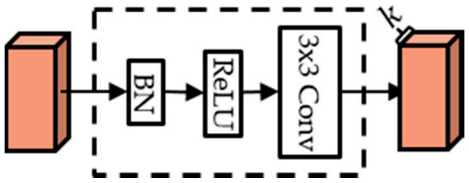


**Figure 2.** Composite function.

In a typical deep neural network, the output of $\ell$-th layer can be expressed as:

$$x_\ell = H_\ell(x_{\ell-1}) \tag{1}$$

In Equation (1), where $x_\ell$ is the output of the $\ell$-th layer, $x_{\ell-1}$ is the output of the $(\ell-1)$-th layer, and it is the input of the $\ell$-th layer. $H_\ell(\cdot)$ is a composite function that is a non-linear transformation.

Deep neural networks have a disadvantage in that they do not learn well when the layer is deep. To overcome this drawback, ResNet [28] used a skip connection whose input is added to the output of the same layer as:

$$x_\ell = H_\ell(x_{\ell-1}) + x_{\ell-1} \tag{2}$$

The skip connection of ResNet allows the gradient to be propagated directly to the previous layers during learning, helping to learn well in deeper architectures. However, as the input and output of the layer are summed, there is a disadvantage that the information in the previous layers becomes weaker.

To overcome the disadvantages of ResNet skip connection above, DenseNet [18] proposes a way to concatenate the feature maps of all preceding layers. Output $x_\ell$ of the $\ell$-th layer is expressed as:

$$x_\ell = H_\ell([x_0, x_1, \ldots, x_{\ell-1}]) \tag{3}$$

where $[x_0, x_1, \ldots, x_{\ell-1}]$ refers to concatenation of the output feature map of layers 0 to $(\ell-1)$. This concatenation scheme is effective for training because the gradient is propagated directly to the previous layers and the input is fed forward directly to the following layers. In the feed-forward process, all of the previous layer outputs are used as input, so that it can compensate that the information of the previous layer becomes weaker as the layer passes. The number of output feature maps for each composite function is denoted by growth rate $k$. In Figure 1, The number of final feature maps of a dense block can be expressed as $m_0 + k \times L$. Here, $m_0$ is the number of feature maps for an input of a dense block, and $L$ is the number of composite functions. When passing through the layer, feature maps are increased by k.

Due to the concatenation characteristics of DenseNet, the number of feature maps increases very much depending on the growth rate, the number of composite functions and dense blocks. To reduce the number of feature maps, compression blocks are added at the back of the dense block. The compression block consists of BN-ReLU-$1 \times 1$ Conv as Figure 3. If the number of the output feature maps of the dense block is $m$, then the number of feature maps created from compression becomes an

$\lfloor \theta m \rfloor$. Compression rate $\theta$ is a value in the range $0 < \theta \leq 1$, and when $\theta = 1$, the number of feature maps for input and output is the same.

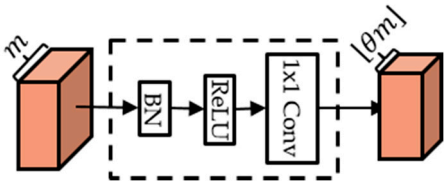


**Figure 3.** Compression block.

### 2.2. Multi-Scale DenseNet for Audio Source Separation

In order to apply DenseNet to audio source separation, the MDenseNet [15] changes input to multi-scale through down-sampling and up-sampling as shown in Figure 4.

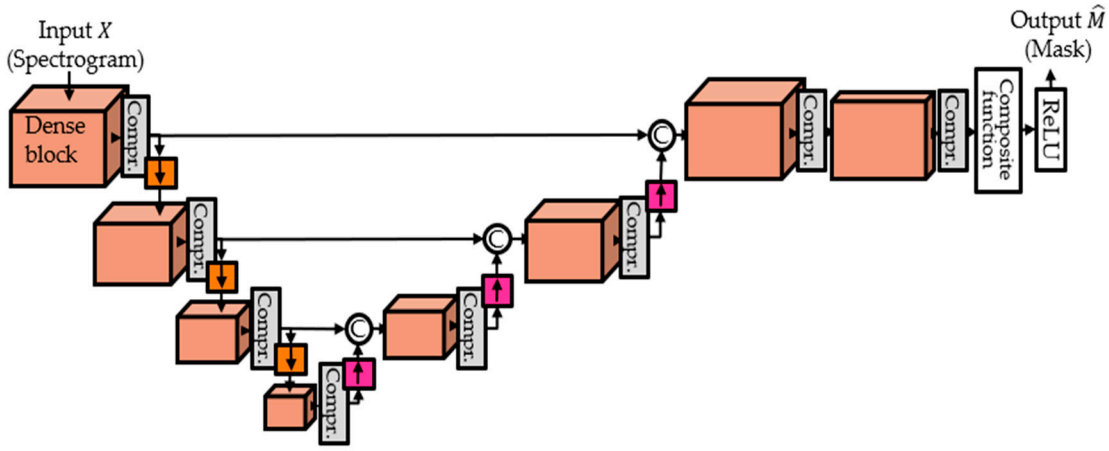


**Figure 4.** MDenseNet architecture.

The down-sampling is done by $2 \times 2$ average pooling. In a classification task, the number of outputs should be equal to the number of classes and, therefore, the input size should be down-sampled in order to match this criterion and there is no up-sampling needed [18]. However, in the task of source separation, the aim is to obtain a mask with the same size as input, therefore, up-sampling stages are needed as well. An up-sampling process is required to restore the down-sampled feature map to the size of the input. For up-sampling, transposed convolution [29] of $2 \times 2$ kernel is used. By down-sampling and up-sampling, it can consider longer context on the time axis and more frequency-range dependency on the frequency axis. In MDenseNet, the feature map output from the dense block of the encoder is connected to the feature map of the same size, which is the input of the dense block in the decoder. Except for the last dense block, the growth rate $k$ and the number of composite functions $L$ are 12 and 4. The $k$ and $L$ of the last dense block is 4 and 2.

## 3. Proposed Architecture for Source Separation

In MDenseNet, down-sampling and up-sampling are undertaken to expand the receptive field. Another way to effectively expand the receptive field is to use dilated convolution [24]. This method showed good performance in a semantic segmentation task [24]. From this motivation, we propose a dilated multi-band multi-scale time-frequency DenseNet architecture. The proposed architecture is shown in Figure 5. Compression rate is 0.25. Down-sampling and up-sampling are undertaken by the same method as the MDenseNet, and also the growth rate $k$ and the number of composite functions $L$ are the same as in MDenseNet [15]. The last composite function is to make the number of feature maps

1, and the last ReLU is to make output values positive because the mask values of the ground truth are all positive. See Appendix A for more details.

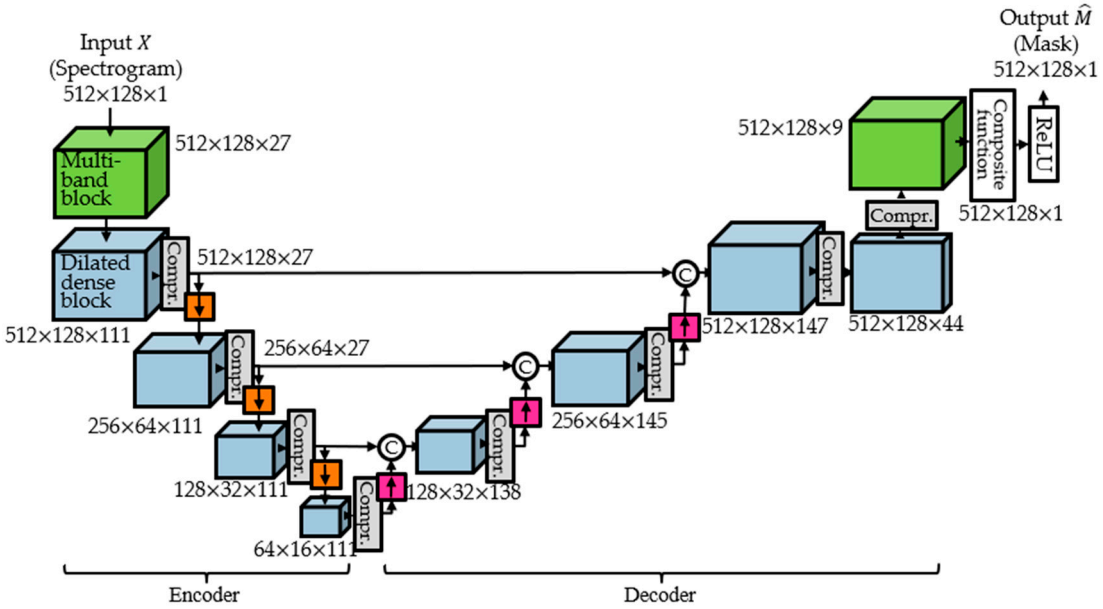


**Figure 5.** Proposed architecture.

### 3.1. Multi-Band Block

The patterns in the spectrogram are different along with the frequency band. The lower frequency band tends to contain high energies, tonalities, and long sustained sounds, while the higher frequency band tends to contain low energies, noise, and rapidly attenuated sound [15]. To reflect these, a multi-band block with a different convolution filter is applied by dividing the spectrogram frequency in half, as shown in Figure 6. *Conv* of the figure is the convolution of $3 \times 3$ kernel. In addition, the entire spectrogram is convolved to obtain a feature map containing information of the entire spectrogram. The final output is obtained by concatenating the two half bands and the full band feature maps. Since we have observed in preliminary experiments that interchanging the order of the feature maps yields improved performance, we interchange the order of the feature maps for low and high bands in a multiband block.

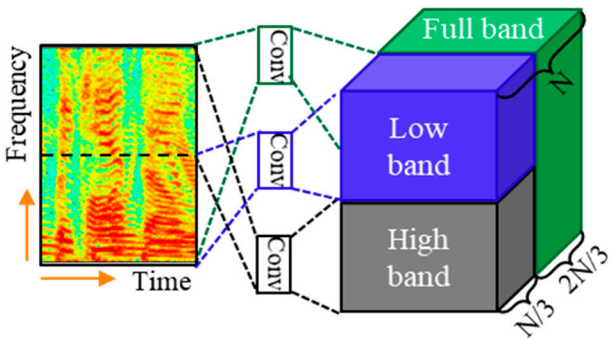


**Figure 6.** Multi-band block.

### 3.2. Dilated Dense Block

The dilated dense block is intended to effectively expand the receptive field. In normal image tasks, the dilation rate of the dilated convolution changes with the equal ratio. However, spectrograms have different characteristics from images in that the time and frequency axes change to different influences. The time axis is affected by the speech rate, and the frequency axis is affected by gender,

pitch, harmonics, and so on. Therefore, the structure of the dilated block is arranged in parallel with time-dilated convolution (TDConv), frequency-dilated convolution (FDConv), and standard convolution, as shown in Figure 7. Also, we experimented with the 2-dilated convolution (2DConv) to compare with TDConv and FDConv. Figure 8 is the kernels of dilated convolution applied to the spectrogram: TDConv, FDConv, and 2DConv. The output of each convolution and the input feature map are concatenated. In the figure, a dilated dense block is represented as the concatenation of a dilated block and a dense block. If the growth rate of this block is $k$, then the feature map output is an integer of $(m_0 + 3 \times k + k \times L)$. Each convolution of the dilated block outputs k feature maps, resulting in the number of feature maps of $3 \times k$. $k \times L$ is the number of output feature maps of the dense block.

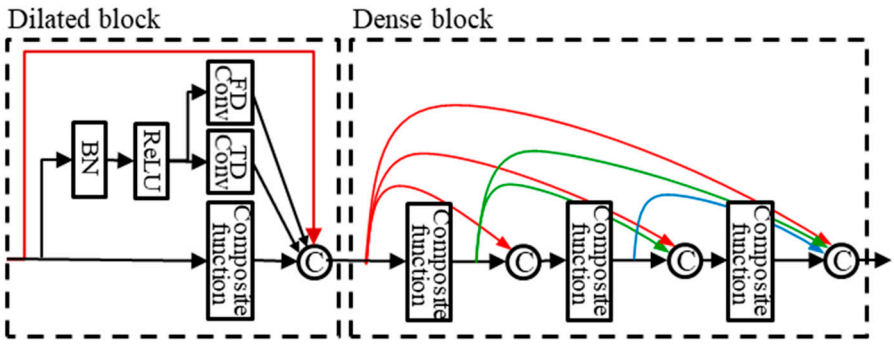


**Figure 7.** Dilated dense block.

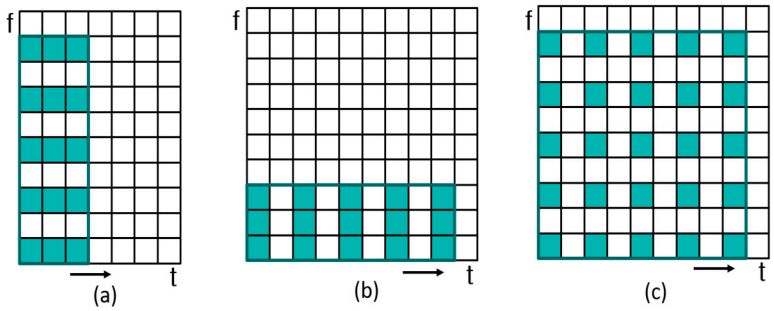


**Figure 8.** Dilated convolution. (**a**) Frequency-dilated convolution, (**b**) time-dilated convolution, (**c**) 2-dilated convolution.

### *3.3. Dropout*

Dropout [30] is applied after each convolution layer of the dilated dense block to prevent overfitting. Dropout is a method of regularization by generating noise to hidden units [30]. If the dropout is applied incorrectly, it does not converge. When we applied the dropout to dilated dense blocks of the encoder part, the proposed architecture did not converge. This is because the noise generated by dropout in dilated dense blocks of the encoder part is intensified through the decoder, which hinders convergence. Therefore, dropout was only applied to dilated dense blocks of the decoder part. In this work, the dropout rate was set to 0.2.

### *3.4. Loss Function*

The learned output of the proposed architecture is the mask to be multiplied by the input spectrogram. The loss function can be expressed by the following equation:

$$\text{Loss} = \left\| Y - X \odot \hat{M} \right\|_1 \tag{4}$$

where the input spectrogram is $X$, the mask estimated in the network is $\hat{M}$, the ground truth spectrogram is $Y$, $\odot$ represents an element-wise multiplication of the matrix, and $\|.\|_1$ is 1-norm which represents the

sum of the absolute values of each element of the matrix. The estimated mask can be represented by $\hat{M} = f(X;\Theta)$, and $f(X;\Theta)$ is the neural network model applied to the input $X$ with parameters $\Theta$. In our study, the model parameters $\Theta_v$ and $\Theta_a$ were individually trained for each source, such as *vocals* and *accompaniment*, respectively.

## 4. Experiments

We first perform a singing voice separation experiment using the open dataset (DSD100 dataset) to guarantee the reproducibility of the experiment. Then, we perform a music identification experiment using our own dataset because there is no available open dataset yet for the purpose of music identification of broadcast content. In the open resource, we evaluate the performance of each block and compare the performance of the proposed architecture with that of the previous architecture. In the music identification after source separation, we calculate the separation and identification performance of the proposed architecture and the previous architectures.

### *4.1. Singing Voice-Separation Experiment*

#### 4.1.1. Dataset

We experimented with the DSD100 dataset made for the 2016 SiSEC [31]. The dataset consists of development and test sets. Each set has 50 songs and was recorded in a stereo environment with a sampling rate of 44.1 kHz. Each song has four music sources (*bass*, *drums*, *other*, *vocals*) and a mixture of sources. In the singing voice separation task, the mixture signal is separated into *vocals* and *accompaniment*.

Several studies in music source separation or vocal instrument separation mixed the instrument signals of the different songs to augment data [12,15,20]. In the DSD100 dataset, we also augmented the training data by mixing the instrument signal of different songs. To balance the other class, the training data was augmented by mixing other signal of different songs and bass, vocals, and drums instrument signal of the same song. Since the duration of the signal is different for each type of music, the signal is adjusted based on the duration of the shortest signal.

#### 4.1.2. Setup

We computed the magnitude $X$ of the mixture spectrogram downsampled at 16 kHz and converted to monophonic to be used as input to the model. The spectrogram is obtained by short-time Fourier transform (STFT) with 1024 window size and 75% overlap. In the network, the mask $\hat{M}$ of the target is estimated, and the estimated target spectrogram is obtained by element-wise multiplication of $X$ and $\hat{M}$. The estimated target signal is restored by taking inverse STFT of the estimated target spectrogram and then performing overlap-add. The estimated separated signal is up-sampled to 44.1 kHz for evaluation.

#### 4.1.3. Separation Results

The experiment was conducted in 8 steps. The first architecture (“A1”) is MDenseNet, which is the baseline. The 2nd architecture (“A2”) is MMDenseNet [15]. The 3rd architecture (“A3”) is MMDenseLSTM [20]. The 4th architecture (“A4”) adds a multi-band block to A1. The 5th architecture (“A5”) replaces the dense block with a dilated dense block containing TDConv and standard convolution. The 6th architecture (“A6”) replaces the TDConv of the A5 with FDConv. The 7th architecture (“A7”) replaces the TDConv of the A5 with 2DConv. The 8th architecture (“A8”) is the proposed architecture which includes both TDConv and FDConv (T-FDConv). We programmed MDenseNet (“A1”), MMDenseNet (“A2”), and MMDenseLSTM (“A3”) by ourselves to operate on monophonic signals. A1, A2, and A3 was to compare the performance in the same environment as the proposed architecture (“A8”).

Table 1 shows the experimental results. Results were measured for *vocals* and *accompaniment* by signal-to-distortion ratio (SDR), signal-to-interference ratio (SIR), and signal-to-artifacts ratio (SAR) [32].

We used median as statistics. A4 performs better than A1 in *vocals* and *accompaniment*, respectively. A5 yielded higher SDR than A4, A6, and A7 in *vocals*, while A7 produced higher SDR than A4, A5, and A6 in *accompaniment*. This shows that *vocals* signals have a variety of patterns along the time axis rather than the frequency axis in the spectrogram, and *accompaniment* signals have a variety of patterns simultaneously along the time and the frequency axes in the spectrogram. After we performed several combinations of experiments with TDConv, FDConv, and 2DConv, we found that the proposed architecture, including TDConv and FDConv, produces the best results. In *vocals*, A8 showed significantly higher SDR than A1~A7. In *accompaniment*, A8 showed significantly higher SDR than A1, A2, A4~A7, and had the same SDR as A3. As a result, A8 improved 0.12 dB SDR over A3 (MMDenseLSTM) in *vocals*. The proposed architecture outperforms MMDenseLSTM in *vocals*.

**Table 1.** Separation performance (dB) on DSD100 dataset.

| **Signal** | **Vocals** | | | **Accompaniment** | | |
|---|---|---|---|---|---|---|
| **Statistics** / **Architecture** | **SDR** | **SIR** | **SAR** | **SDR** | **SIR** | **SAR** |
| A1 (MDenseNet) | 4.93 | 12.98 | 6.22 | 11.48 | 16.74 | 13.45 |
| A2 (MMDenseNet) | 5.61 | 13.43 | 6.57 | 11.97 | 16.84 | 13.96 |
| A3 (MMDenseLSTM) | 5.78 | 13.57 | 6.87 | **12.03** | 16.44 | **14.22** |
| A4 (A1 + MB) | 5.31 | 13.51 | 6.30 | 11.62 | 16.24 | 13.38 |
| A5 (A3 + TDConv) | 5.51 | 14.27 | 6.47 | 11.73 | 16.86 | 13.50 |
| A6 (A3 + FDConv) | 5.26 | 13.50 | 6.48 | 11.71 | **17.82** | 13.21 |
| A7 (A3 + 2DConv) | 5.46 | 13.82 | 6.42 | 11.81 | 17.67 | 13.07 |
| A8 (A3 + T-FDConv) | **5.90** | **14.56** | **6.90** | **12.03** | 17.16 | 13.48 |

We compared the proposed architecture with the previous methods: Deep non-negative matrix factorization (DeepNMF), FNN, BLSTM, 4-stacked hourglass network (SH-4stack), MMDenseNet, and MMDenseLSTM. For DeepNMF [9], FNN [8], SH-4stack [14], and the proposed architecture, monophonic signals were used for evaluation. On the other hand, stereo signals were used to evaluate BSLTM [12], MMDenseNet [15], and MMDenseLSTM [20] with data augmentation and multi-channel Wiener filter (MWF). The performance of MMDenseNet [15] and MMDenseLSTM [20] presented in Table 2 differs from our experimental results A2 and A3 in Table 1. This difference is because our experiment used monophonic signals and did not use data augmentation and MWF.

**Table 2.** Signal-to-distortion ratio (SDR, dB) comparison with other methods.

| **Method** | **Number of Parameters ($\times 10^6$)** | **Vocals** | **Accompaniment** |
|---|---|---|---|
| DeepNMF [9] | - | 2.75 | 8.90 |
| FNN [8] | - | 4.47 | 11.12 |
| BLSTM [12] | 30.03 | 4.86 | 11.26 |
| SH-4stack [14] | 34.18 | 5.45 | 12.14 |
| MMDenseNet [15] | 0.33 | 6.00 | 12.10 |
| MMDenseLSTM [20] | 1.22 | **6.31** | **12.73** |
| Our method | 0.48 | 6.25 | 12.58 |

To obtain the results of our method in Table 2, we also applied data augmentation techniques to the A8 architecture for comparing performance with the previous methods. Here, we augmented training data by using three times the original data. The proposed method has higher SDR than BLSTM, therefore we can see that CNN-based architectures outperform BLSTM. The proposed method using DenseNet, which improves information flow between layers or blocks, outperforms a typical CNN-based SH-4 stack. The proposed method is better than MMDenseNet by effectively expanding the receptive field. The proposed method shows lower performance than MMDenseLSTM, but it does not use the Wiener filter and has the advantage of network configuration with very few parameters.

As a result, the proposed method, which effectively expands the receptive field at DenseNet, showed the highest performance next to the MMDenseLSTM with 6.25 dB SDR in *vocals* and 12.58 dB SDR in *accompaniment*.

### 4.2. Music Identification Experiment

The music identification experiment has the structure shown in Figure 9. The separation model is trained using mixed music and speech signals as inputs and each original signals as a reference in the training process. In the test process, the mixed signal is separated into the speech and music signal by the trained each separation model, and fingerprinting features are extracted from the separated music signal. The separated speech signal is only used to measure the separation performance. Music identification is performed in the fingerprinting database using the landmark-based fingerprinting feature [33] of the separated music signal. We can obtain the identification result, and we can also calculate the separation performance from the separated signal.

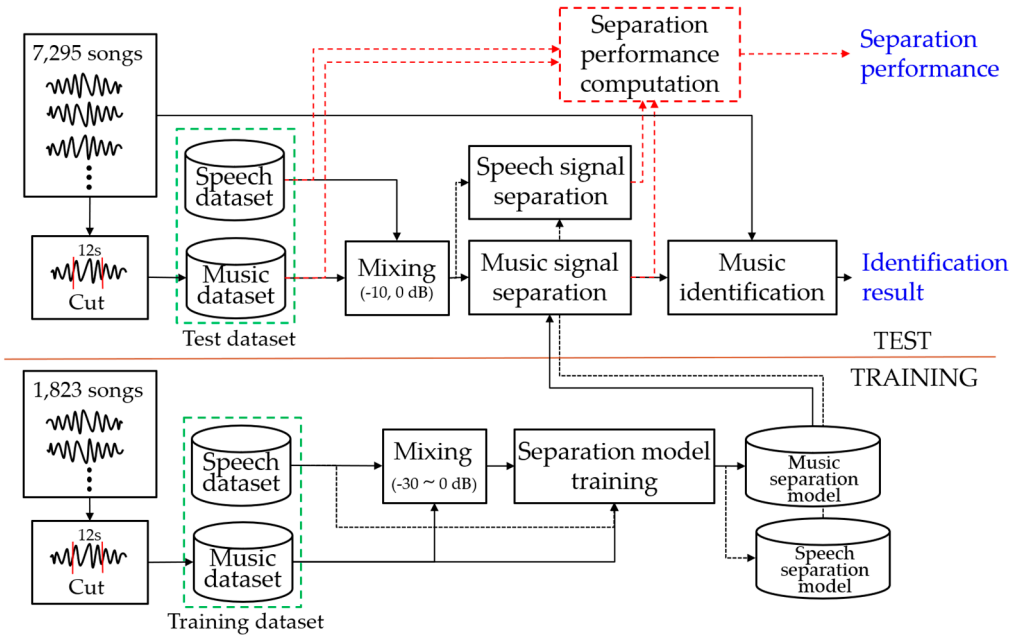


**Figure 9.** Overall structure of music identification experiment.

#### 4.2.1. Dataset

For the music identification experiment, 9118 songs of various countries and genres were collected; 1823 songs were used for the training and 7295 songs were used for the test. Landmark information was extracted from all sections of 7295. The length of the query signal was 12 s.

The speech data was extracted only in a section where only pure speech exists in 90 h of broadcast content of various genres. The duration per genre of broadcast content was 30 h each for drama and entertainment, and 15 h each for documentaries and kids. The extracted speech signals were divided into 12-s intervals to generate a total of 3646 speech data corresponding to about 12 h; 1823 speech samples were used for the training and the remaining 1823 speech samples were used for the test. The speech data was recorded in a stereo format at 44,100 Hz sampling rate.

The music signal of the training data was cut to 12 s in an arbitrary section. The truncated music signal was mixed with the speech signal to have an arbitrary signal-to-noise ratio (SNR) between –30 and 0 dB to apply the characteristics of the broadcaster where the louder speech signal was mixed. The music signal of the test data was cut off for 12 s in two arbitrary sections of each song. The truncated 12-s music signal was mixed with the test speech data and tested. In the test, the SNR of 0 dB and −10

dB were mixed to measure the separation performance according to each SNR. There are 14,590 test query data for each SNR. The music signal was recorded in a stereo format at a 44,100 Hz sampling rate.

#### 4.2.2. Mixing

In order to create a dataset with an environment similar to broadcast contents, music signals and speech signals should be mixed to appropriate SNRs. The voice activity detection (VAD) was used to find the section in which the actual speech exists to blend to the desired SNR. The equation below is for creating mixed data:

$$\alpha = 10^{\frac{\beta}{20}} \times \frac{P_{avg}(x_s[v_s])}{P_{avg}(x_m[v_m])} \tag{5}$$

$$y = x_s + \alpha \times x_m \tag{6}$$

where $\alpha$ is a mixing factor with the target SNR $\beta$ (dB), $x_s$ is a speech signal, $x_m$ is a music signal, and $P_{avg}(\cdot)$ is average power. The $v_s$ and $v_m$ is an output vector of WebRTC [34] and represents a vector where actual speech signal and music signal are located, respectively. The $y$, which is the mixed signal, can be obtained by linearly adding $x_m$ multiplied by $\alpha$ and $x_s$.

#### 4.2.3. Setup

We used mono signals down-sampled at 16 kHz and calculated spectrograms with 1024 window size and 75% overlap size. The output spectrogram of the separation system was converted and saved as a waveform, which was used as the input for music identification. For music identification, we used the open-source landmark-based identification program [35] and we experimented with the default values set in the program. Wave-U-Net experiments were conducted using the open source program [36].

#### 4.2.4. Music Identification Results

In order to fairly compare with other methods, we did not use data augmentation or Wiener filter for the separation system. Table 3 shows the SDR, SIR, and SAR performance of the separated music and speech signals by each separation architecture.

**Table 3.** Separation results (dB) on broadcast contents.

| **Separated Signal** | **Music** | | | | | | **Speech** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **SNR** | **0 dB** | | | **−10 dB** | | | **0 dB** | | | **−10 dB** | | |
| **Statistics / Architecture** | **SDR** | **SIR** | **SAR** | **SDR** | **SIR** | **SAR** | **SDR** | **SIR** | **SAR** | **SDR** | **SIR** | **SAR** |
| U-Net | 6.24 | 11.51 | 8.54 | 3.18 | 10.72 | 4.63 | 5.66 | 9.10 | 9.36 | 13.65 | 16.97 | 17.04 |
| Wave-U-Net | 6.33 | **14.19** | 7.54 | 2.97 | **13.15** | 3.70 | 6.02 | 10.32 | 8.97 | 14.24 | 18.70 | 16.66 |
| MDenseNet | 6.98 | 13.23 | 8.63 | 3.84 | 11.32 | 5.14 | 6.57 | 10.80 | 9.42 | 14.45 | 18.63 | 17.01 |
| MMDenseNet | 7.15 | 13.43 | 8.86 | 4.04 | 11.48 | 5.35 | 6.77 | 10.92 | 9.58 | 14.53 | 18.74 | 17.07 |
| MMDenseLSTM | 7.40 | 13.34 | 9.22 | 4.13 | 11.21 | 5.54 | **7.68** | **12.36** | **10.02** | 14.98 | **19.41** | 17.33 |
| Proposed | 7.72 | 13.74 | **9.45** | **4.44** | 11.72 | 5.72 | 7.63 | 12.29 | 9.98 | **15.00** | 19.39 | **17.34** |

In the music separation result, Wave-U-Net performs best in SIR performance. SIR is a quantification of the degree of interference between speech and music signals, which shows how much speech signal remains in the separated music signal. Other structures except Wave-U-Net estimate the spectrogram magnitude of the music and reconstruct the signal using the phase information of the mixed signal. Using the phase information of the mixed signal causes the SIR to be lowered. To avoid this interference, Wave-U-Net estimates the signal directly in the time domain. However, in terms of SDR, MDenseNet, MMDenseNet, MMDenseLSTM, and the proposed architecture using DenseNet show higher performance than Wave-U-Net. Since SDR is a comprehensive value considering both SIR and SAR, separation performance is usually compared based on SDR. The proposed architecture

showed the best separation performance as 7.72 dB SDR at 0 dB SNR and 4.44 dB SDR at −10 dB SNR. At −10 dB SNR, Wave-U-Net showed lower separation performance than U-Net. Except for this case, the performance of each architecture showed reasonable results, as can be seen in previous and other studies [19,37].

In the speech separation result, we can see that unlike music, the SIR performance of Wave-U-Net is lower than that of DenseNet-based separation architectures. the speech signal separation is less affected by the phase of the mixed signal than the music signal separation. We can see that the separation performance of MMDenseLSTM and the proposed architecture is similar and is better than other architectures. However, in music identification, the proposed architecture outperformed MMDenseLSTM.

Table 4 shows the accuracy of music identification. The performance of the *Mix* is the lower boundary, and the performance of the *Oracle* is the upper boundary. The performance of the *Oracle* is not 100% because of the distortion caused by the down-sampling. U-Net showed the lowest identification performance, and the proposed architecture showed the best performance in identification with 71.91% identification accuracy at 0 dB SNR and 48.03% identification accuracy at −10 dB SNR. However, some results did not correlate with SDR performance. MMDenseNet had higher separation performance than Wave-U-Net and MDenseNet, but lower identification performance.

Figure 10 shows the music identification results and the fingerprinting features at the spectrogram for the query signal of each separation system. Fingerprinting features appear up to 5512 Hz by the setting of the identification system. These figures show that MDenseNet and MMDenseNet are poorly identified despite high SDR. In *Result 1* of the figure, the separated signal by MMDenseNet with the second-highest SDR fails to be identified. However, the proposed architecture with the same SDR performance as MMDenseNet is successfully identified. In *Result 2* of the figure, identification is successful although MDenseNet produces lower SDR than MMDenseNet. In contrast, MMDenseNet has the highest SDR but fails to be identified.

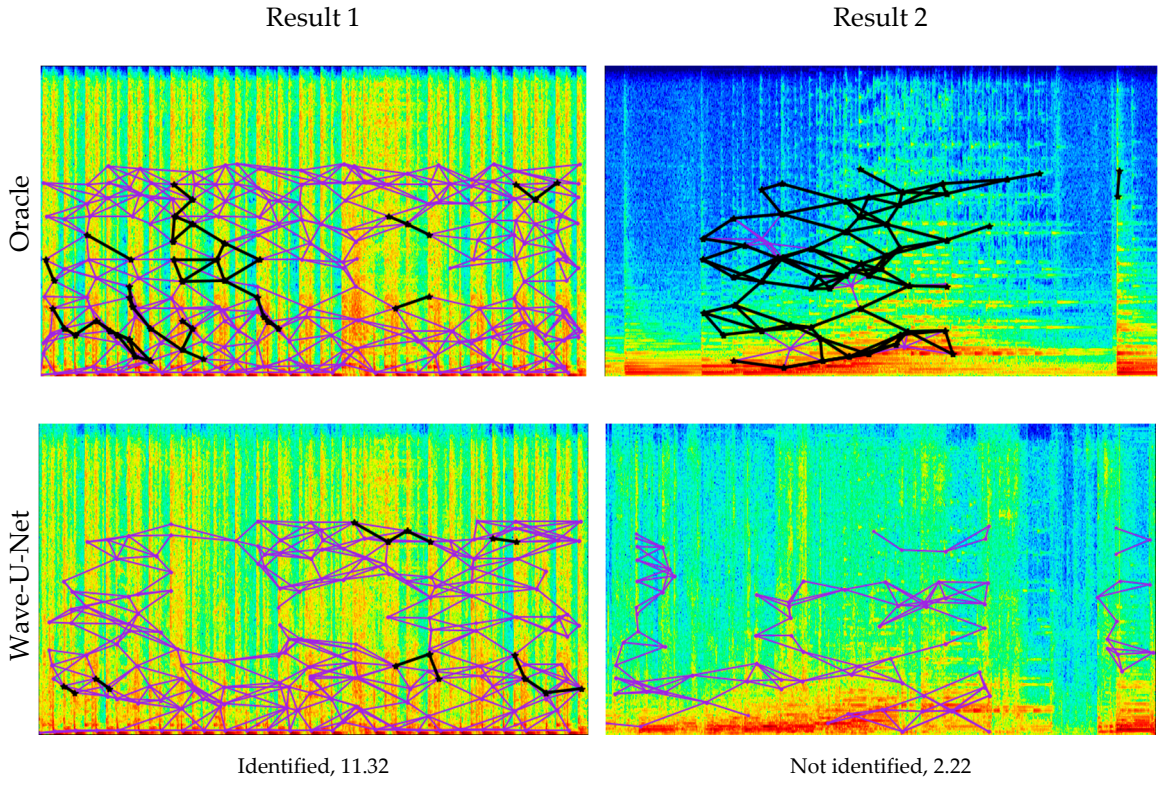


**Figure 10.** *Cont.*

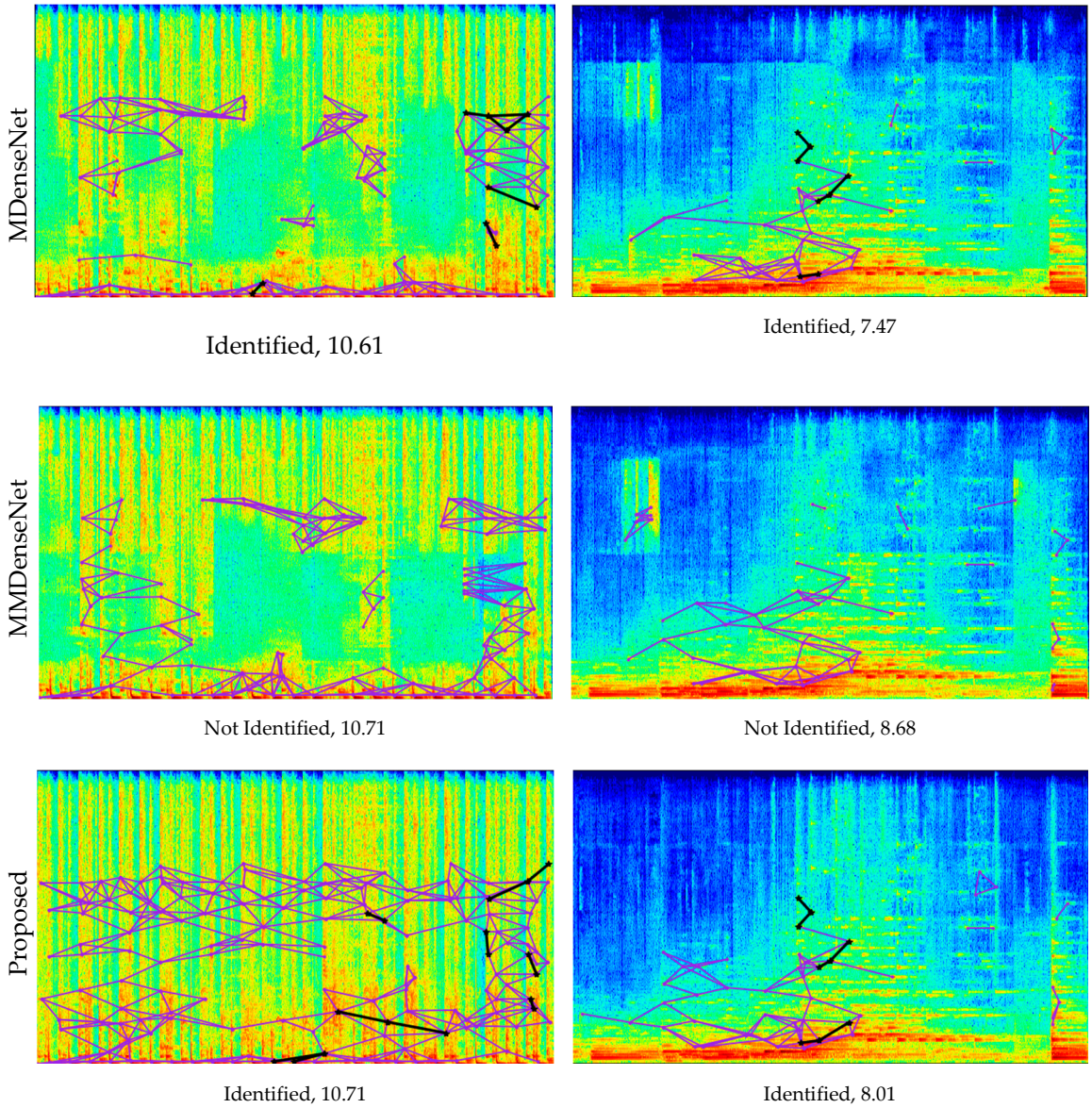


**Figure 10.** Music identification results and fingerprinting plots at spectrogram. The horizontal axis of the spectrogram represents time (0~12 s) and the vertical axis represents frequency (0~8000 Hz). The matching result and the SDR of the signal are displayed under each plot. The violet line is the fingerprinting of the query, and the black line is the fingerprinting that matches the reference.

**Table 4.** Music identification accuracy (%).

| Architecture \ SNR | 0 dB | −10 dB |
|---|---|---|
| Mix | 43.38 | 4.59 |
| U-Net | 42.33 | 19.38 |
| Wave-U-Net | 54.77 | 26.89 |
| MDenseNet | 52.57 | 28.44 |
| MMDenseNet | 49.66 | 26.67 |
| MMDenseLSTM | 67.94 | 44.96 |
| Proposed | **71.91** | **48.03** |
| Oracle | 95.92 | |

At the spectrogram in Figure 10, we can see discontinuous horizontal lines in MDenseNet and MMDenseNet. There is a discontinuous horizontal line at the 6500 Hz frequency of the MDenseNet spectrogram. This distortion is not frequent, and the discontinuous horizontal line appears at 6500 Hz in MMDenseNet. In addition, MMDenseNet notices an additional discontinuous horizontal line in the middle (4000 Hz) of the spectrogram. Placing an excessive number of parameters in each frequency band in parallel, such as MMDenseNet, is not effective for increasing the receptive field and introduces distortion on the spectrogram. This distortion of the spectrogram has a small effect on SDR but is an obstacle to extract for fingerprinting features in music identification. Designing to apply convolution to each band of the spectrogram like the multi-band block of the proposed structure can prevent distortion and effectively increase the receptive field. In the spectrogram of the proposed architecture, we can see that no distortion occurs. Even with these discontinuous horizontal lines, the separation performance of MMDenseNet is higher than that of MDenseNet because the SDR is sensitive to the low-frequency band of the spectrogram. Additional experiments in this regard are covered in the next subsection.

The high identification performance of Wave-U-Net is expected to be related to the receptive field. A large receptive field of the network is advantageous for maintaining the peak points of the estimated spectrogram. Wave-U-Net has 12 down-sampling processes in the signal domain. The proposed architecture that effectively increases the receptive field by dilated convolution and down-sampling shows the best performance in identification as well as high SDR.

#### 4.2.5. Signal-to-Distortion Ratio (SDR) Comparison of Distortion in the Spectrogram Frequency Band

We experimented to see how the frequency band distortion of the spectrogram affects SDR. As shown in Figure 11, the spectrogram of music signals was divided into four bands (*D1~D4*) and multiplied by the weight for each band to distort it. The music signal without distortion is called *D0*. We added the distorted music spectrogram to the speech spectrogram and restored it to the signal. Finally, the SDR was calculated using the distorted mixture signal and the reference signals with 16 kHz sampling rate. In addition, the identification experiment was performed using the distorted mixed signal. The fingerprinting database was extracted at 16 kHz sampling rate from 7295 songs. We used the distortion weight $\alpha$ as 0.3. The 1000 samples mixed with 0 dB SNR were tested.

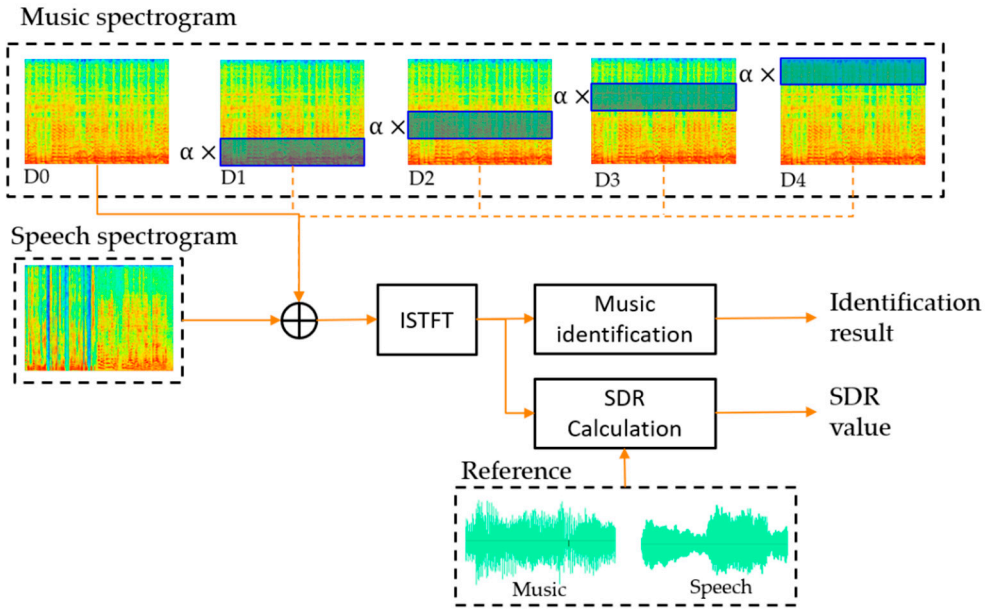


**Figure 11.** Experimental structure of SDR measurement according to distortion by frequency band.

Table 5 shows the SDR and identification results of the distorted mixed signal. The mean SDR of the undistorted signal *D0* matches the mixed SNR and shows the best identification accuracy. *D1*

shows the lowest SDR value. On the other hand, *D1* shows better identification accuracy than *D2~D4*. *D2~D4* show higher SDR performance than *D1*, but lower identification accuracy than *D1*.

**Table 5.** SDR and identification result according to frequency band distortion.

| Distortion | Mean SDR (dB) | Identification Accuracy (%) |
|---|---|---|
| D0 | −0.01 | 48.9 |
| D1 | −8.53 | 42.9 |
| D2 | −0.30 | 40.9 |
| D3 | −0.10 | 40.6 |
| D4 | −0.05 | 40.9 |

Since the SDR of separation performance is calculated based on the correlation of signal domain, it is sensitive to the low-frequency band with high energy values. A high SDR performance can be obtained by accurately estimating the low-frequency portion of the spectrogram. However, since speech signal has higher energy in the low-frequency band than music signal, maintaining the peak points of the high-frequency band at the music signal is important to music identification. For this reason, the identification performance is lower even with high SDR.

## 5. Conclusions

In this study, we proposed source separation using dilated time-frequency DenseNet for music identification in broadcast content. The background music of broadcast content is frequently mixed with speech, and further the volume of music signal is less than the volume of the speech signal in most cases. In this case, music identification is not easy, and hence background music separation is required before music identification.

In previous studies, source separation using deep learning was studied extensively and showed a good performance. We add a time-frequency dilated convolution and apply different convolutions to each frequency band of the spectrogram to effectively increase the receptive field in the CNN-based DenseNet architecture. We conducted a music-identification experiment by separating the music signal from mixture signals into the proposed architecture and the previous architecture.

The results of music identification did not correlate with the separation performance. Wave-U-Net, MDenseNet, and MMDenseNet results of the music identification experiments were in contrast to the separation performance. The separation performance of SDR was affected by the low-frequency region of the spectrogram. The music identification module extracted the fingerprinting feature using the peak points of the spectrogram. Accordingly, if only the peak points of the separated signal are well preserved, the identification is likely to succeed. Despite the different characteristics in performance, the proposed architecture showed the best performance in identification as well as in separation.

**Author Contributions:** Data curation, W.-H.H. and H.K.; Conceptualization, W.-H.H.; Methodology, W.-H.H. and O.-W.K.; Validation, W.-H.H. and H.K.; Writing-original draft preparation, W.-H.H.; Writing-review and editing, O.-W.K.; Supervision, O.-W.K.; all of the authors participated in the project, and they read and approved the final manuscript. All authors have read and agreed to the published version of the manuscript

**Funding:** This research received no external funding.

**Acknowledgments:** This research project was supported by Ministry of Culture, Sports and Tourism (MCST) and from Korea Copyright Commission in 2020. [2018-micro-9500, Intelligent Micro-Identification Technology for Music and Video Monitoring]

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

Table A1 shows the proposed architecture in detail. In the table, $f \times t$ of the multi-band block is the kernel size of convolution, and N is the number of output feature maps. In the dilated dense

block, $k$ is the growth rate, $L$ is the number of composite layers, $\epsilon$ is the dropout rate, and $\theta$ is the compression rate.

**Table A1.** Details of the proposed architecture.

| Layers | Parameter | Output Size |
|---|---|---|
| Multi-band 1 ($f \times t$, N) | 3×3, 27 | 512×128×27 |
| Dilated Dense 1 ($k$, $L$, $\epsilon$) | 12, 4, 0 | 512×128×111 |
| Compression ($\theta$) | 0.25 | 512×128×27 |
| Down-sampling | 2×2 average pooling | 256×64×27 |
| Dilated Dense 2 ($k$, $L$, $\epsilon$) | 12, 4, 0 | 256×64×111 |
| Compression ($\theta$) | 0.25 | 256×64×27 |
| Down-sampling | 2×2 average pooling | 128×32×27 |
| Dilated Dense 3 ($k$, $L$, $\epsilon$) | 12, 4, 0 | 128×32×111 |
| Compression ($\theta$) | 0.25 | 128×32×27 |
| Down-sampling | 2×2 average pooling | 64×16×27 |
| Dilated Dense 4 ($k$, $L$, $\epsilon$) | 12, 4, 0 | 64×16×111 |
| Compression ($\theta$) | 0.25 | 64×16×27 |
| Up-sampling | 2×2 transposed convolution | 128×32×27 |
| Concatenate | Dilated Dense 3 | 128×32×54 |
| Dilated Dense 5 ($k$, $L$, $\epsilon$) | 12, 4, 0.2 | 128×32×138 |
| Compression ($\theta$) | 0.25 | 128×32×34 |
| Up-sampling | 2×2 transposed convolution | 256×64×34 |
| Concatenate | Dilated Dense 2 | 256×64×61 |
| Dilated Dense 6 ($k$, $L$, $\epsilon$) | 12, 4, 0.2 | 256×64×145 |
| Compression ($\theta$) | 0.25 | 256×64×36 |
| Up-sampling | 2×2 transposed convolution | 512×128×36 |
| Concatenate | Dilated Dense 1 | 512×128×63 |
| Dilated Dense 7 ($k$, $L$, $\epsilon$) | 12, 4, 0.2 | 512×128×147 |
| Compression ($\theta$) | 0.25 | 512×128×36 |
| Dilated Dense 8 ($k$, $L$, $\epsilon$) | 4, 2, 0.2 | 512×128×44 |
| Compression ($\theta$) | 0.25 | 512×128×11 |
| Multi-band 2 ($f \times t$, N) | 3×3, 9 | 512×128×9 |
| BN-ReLU | - | (same) |
| Conv ($f \times t$, N) | 3×3, 1 | 512×128×1 |
| ReLU | - | (same) |

## References


1. Vincent, E.; Bertin, N.; Gribonval, R.; Bimbot, F. From blind to guided audio source separation: How models and side information can improve the separation of sound. *IEEE Signal Process. Mag.* **2014**, *31*, 107–115. [CrossRef]
2. Hyvärinen, A.; Oja, E. Independent component analysis: Algorithms and applications. *Neural Netw.* **2000**, *13*, 411–430. [CrossRef]
3. Lee, D.D.; Seung, H.S. Algorithms for non-negative matrix factorization. In Proceedings of the Neural Information Processing Systems (NIPS), Vancouver, BC, Canada, 3–8 December 2001; pp. 556–562.
4. Georgiev, P.; Theis, F.; Cichocki, A. Sparse component analysis and blind source separation of underdetermined mixtures. *IEEE Trans. Neural Netw.* **2005**, *16*, 992–996. [CrossRef] [PubMed]
5. Simpson, A.J.; Roma, G.; Plumbley, M.D. Deep karaoke: Extracting vocals from musical mixtures using a convolutional deep neural network. In Proceedings of the International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA), Liberec, Czech Republic, 25–28 August 2015; pp. 429–436.
6. Grais, E.; Sen, M.; Erdogan, H. Deep neural networks for single channel source separation. In Proceedings of the IEEE International Conference on Acoustics, Speech, Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 3734–3738.
7. Nugraha, A.A.; Liutkus, A.; Vincent, E. Multichannel music separation with deep neural networks. In Proceedings of the European Signal Processing Conference (EUSIPCO), Budapest, Hungary, 29 August–2 September 2015; pp. 1748–1752.

8. Uhlich, S.; Giron, F.; Mitsufuji, Y. Deep neural network based instrument extraction from music. In Proceedings of the IEEE International Conference on Acoustics, Speech, Signal Processing (ICASSP), Brisbane, Australia, 19–24 April 2015; pp. 2135–2139.
9. Le Roux, J.; Hershey, J.; Weninger, F. Deep NMF for speech separation. In Proceedings of the IEEE International Conference on Acoustics, Speech, Signal Processing (ICASSP), Brisbane, Australia, 19–24 April 2015; pp. 66–70.
10. Sawada, H.; Mukai, R.; Araki, S.; Makino, S. A robust and precise method for solving the permutation problem of frequency-domain blind source separation. *IEEE Trans. Speech Audio Process.* **2004**, *12*, 530–538. [CrossRef]
11. Huang, P.-S.; Kim, M.; Hasegawa-Johnson, M.; Smaragdis, P. Deep learning for monaural speech separation. In Proceedings of the IEEE International Conference on Acoustics, Speech, Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 1562–1566.
12. 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 Proceedings of the IEEE International Conference on Acoustics, Speech, Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 261–265.
13. Jansson, A.; Humphrey, E.; Montecchio, N.; Bittner, R.; Kumar, A.; Weyde, T. Singing voice separation with deep U-Net convolutional networks. In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Montreal, MT, Canada, 4–8 November 2017; pp. 745–751.
14. Park, S.; Kim, T.; Lee, K.; Kwak, N. Music source separation using stacked hourglass networks. In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Paris, France, 23–27 September 2018; pp. 289–296.
15. Takahashi, N.; Mitsufuji, Y. Multi-scale multi-band DenseNets for audio source separation. In Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz, NY, USA, 15–18 October 2017; pp. 21–25.
16. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Munich, Germany, 5–9 October 2015; pp. 234–241.
17. Newell, A.; Yang, K.; Deng, J. Stacked hourglass networks for human pose estimation. In Proceedings of the European Conference Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 483–499.
18. Huang, G.; Liu, Z.; Weinberger, K.Q.; Maaten, L. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 24–30 June 2017; pp. 4700–4708.
19. Stoller, D.; Ewert, S.; Dixon, S. Wave-u-net: A multi-scale neural network for end-to-end audio source separation. *arXiv* **2018**, arXiv:1806.03185.
20. Takahashi, N.; Goswami, N.; Mitsufuji, Y. MMDenseLSTM: An efficient combination of convolutional and recurrent neural networks for audio source separation. In Proceedings of the International Workshop on Acoustic Signal Enhancement (IWAENC), Tokyo, Japan, 17–20 September 2018; pp. 106–110.
21. Takahashi, N.; Agrawal, P.; Goswami, N.; Mitsufuji, Y. PhaseNet: Discretized phase modeling with deep neural networks for audio source separation. In Proceedings of the INTERSPEECH, Hyderabad, India, 2–6 September 2018; pp. 2713–2717.
22. Hubel, D.H.; Wiesel, T.N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* **1962**, *160*, 106–154. [CrossRef] [PubMed]
23. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
24. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. In Proceedings of the International Conference on Learning Representations (ICLR), San Juan, PR, USA, 2–4 May 2016.
25. Jang, B.Y.; Heo, W.H.; Kim, J.H.; Kwon, O.W. Music detection from broadcast contents using convolutional neural networks with a Mel-scale kernel. *EURASIP J. Audio Speech Music Process.* **2019**, *2019*, 11. [CrossRef]
26. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015; pp. 448–456.

27. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
29. Dumoulin, V.; Visin., F. A guide to convolution arithmetic for deep learning. *arXiv* **2016**, arXiv:1603.07285.
30. Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
31. Liutkus, A.; Stöter, F.; Rafii, Z.; Kitamura, D.; Rivet, B.; Ito, N.; Ono, N.; Fontecave, J. The 2016 signal separation evaluation campaign. In Proceedings of the International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA), Grenoble, France, 21–23 February 2017; pp. 66–70.
32. Vincent, E.; Gribonval, R.; Fevotte, C. Performance measurement in blind audio source separation. *IEEE Trans. Audio Speech Lang. Process.* **2006**, *14*, 1462–1469. [CrossRef]
33. Wang, A. An industrial strength audio search algorithm. In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Baltimore, MD, USA, 26–30 October 2003; pp. 7–13.
34. WebRTC. Available online: https://github.com/wiseman/py-webrtcvad (accessed on 20 September 2019).
35. Audfpint. Available online: https://github.com/dpwe/audfprint (accessed on 20 September 2019).
36. Wave-U-Net. Available online: https://github.com/f90/Wave-U-Net (accessed on 4 December 2019).
37. Ward, D.; Mason, R.D.; Kim, C.; Stöter, F.R.; Liutkus, A.; Plumbley, M. SiSEC 2018: State of the art in musical audio source separation-subjective selection of the best algorithm. In Proceedings of the Workshop on Intelligent Music Production, Huddersfield, UK, 14 September 2018.