國立臺灣大學電機資訊學院資訊網路與多媒體研究所
碩士論文

Graduate Institute of Networking and Multimedia
College of Electrical Engineering and Computer Science
National Taiwan University
Master Thesis

使用多重擴張卷積 MMDenseNet 於即時歌曲伴奏分離

Real-Time Accompaniment Extraction with Multi-Dilated Convolution MMDenseNet

李學翰

Hsueh-Han Lee

指導教授：張智星　博士

Advisor: Jyh-Shing Roger Jang, Ph.D

中 華 民 國 112 年 8 月

Aug, 2023

doi:10.6342/NTU202301173

doi:10.6342/NTU202301173

國立臺灣大學碩士學位論文
口試委員會審定書

使用多重擴張卷積 MMDenseNet 於即時歌曲伴奏分離

Real-Time Accompaniment Extraction with Multi-Dilated Convolution MMDenseNet

本論文係李學翰君（學號 R10944042）在國立臺灣大學資訊網路與多媒體研究所完成之碩士學位論文，於民國一百一十二年七月三十日承下列考試委員審查通過及口試及格，特此證明。

口試委員：張智星 （簽名）

（指導教授）

蘇黎　　曹昱

所　長：鄭卜壬

doi:10.6342/NTU202301173

使用多重擴張卷積 MMDenseNet 於即時歌曲伴奏分離

摘要

「音樂聲部分離」為音樂資訊檢索領域中重要研究方向，其目標為將一由多部聲源混合而成之音樂訊號，還原回各自混合前的訊號。而音樂聲部分離的子任務「歌曲人聲分離」，則致力於將音樂訊號還原為「人聲」和「伴奏」兩個音軌，即使已有許多研究提出架構達到良好的分離效果，卻都伴隨相當龐大的運算資源與時間，並不適用於即時分離系統的應用，因此如何即時進行伴奏音軌的分離，即為本文研究方向。本文使用音樂聲部分離領域中一輕量模型架構 MMDenseNet，先以遮罩預測、多重擴張卷積、增加模型複雜度等方式提升分離效果，再以縮短模型輸入長度和上下文聚合等方式降低延遲時間，以達到擁有良好分離效果且低延遲之模型。

關鍵字： 音樂聲部分離、歌曲人聲分離、MMDenseNet、多重擴張卷積、頻譜遮罩預測、即時分離

doi:10.6342/NTU202301173

doi:10.6342/NTU202301173

Real-Time Accompaniment Extraction with Multi-Dilated Convolution MMDenseNet

Abstract

Music source separation (MSS) is an important research task in the music information retrieval (MIR) domain which aims to recover the mixing of musical signals to individual audio tracks. And its subtask, singing voice separation (SVS), is dedicated to recovering the signal to vocals and accompaniment tracks merely. Although several studies proposed their methods to achieve outstanding performances, the massive computing power and processing time limit the applications on edge devices. Therefore, extracting the accompaniment track in real-time with limited resources is the main target in this article. A lightweight MSS model, MMDenseNet, is used in this study. With mask estimation, multi-dilated convolution, and model complexity increasing, the separation performance is enhanced. And with shorter model input duration and context aggregation, the latency is decreased. Therefore the separation can be performed in real time and the performance is sustained.

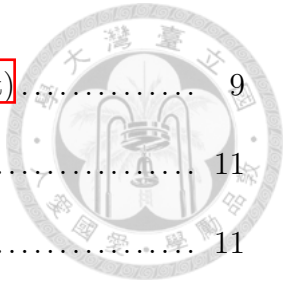
Keywords: Music Source Separation, Singing Voice Separation, MMDenseNet, Multi-dilated Convolution, Spectral mask estimation, Real-time separation

doi:10.6342/NTU202301173

doi:10.6342/NTU202301173

Contents

doi:10.6342/NTU202301173



doi:10.6342/NTU202301173

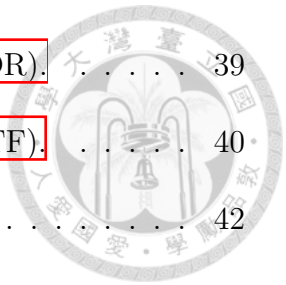
doi:10.6342/NTU202301173

doi:10.6342/NTU202301173

List of Figures

page

doi:10.6342/NTU202301173

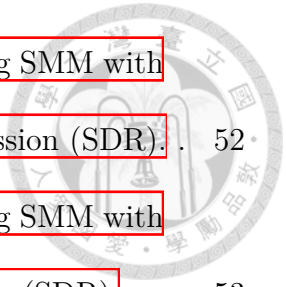


doi:10.6342/NTU202301173

List of Tables

page

doi:10.6342/NTU202301173



doi:10.6342/NTU202301173

Chapter 1

Introduction

1.1 Singing voice separation (SVS)

Singing voice separation (SVS) is a critical topic in the scope of musical information retrieval (MIR). It is a subproblem of another topic, music source separation (MSS). While MSS targets reconstructing the signal of each instrument from the mixture, SVS focuses on extracting vocals and accompaniment tracks only.

The technique of SVS can be used in several applications. For example, the accompaniment extraction module can be incorporated into the karaoke system, the separation of vocals can be a preprocessing to another downstream task, such as melody extraction and transcription, and both the vocals and accompaniment parts can be used in music remixing.

doi:10.6342/NTU202301173

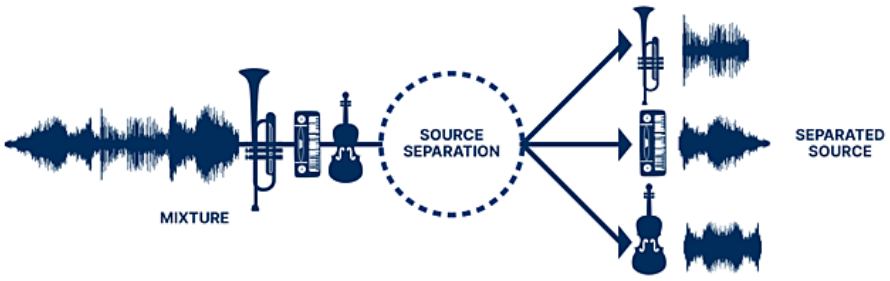


Figure 1.1: Music source separation [1]

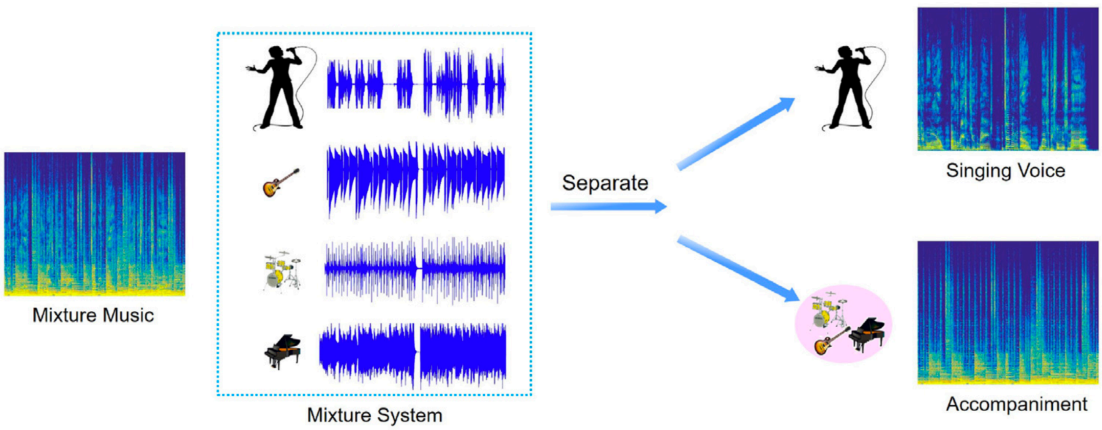


Figure 1.2: Singing Voice separation [2]

doi:10.6342/NTU202301173

1.2 Research topic

In the past, most of the traditional methods dealt with the task by performing matrix-based operations in the spectral domain. As the deep learning technique became popular, several research institutes and organizations proposed their deep neural networks (DNN) based solutions to the problem and outperformed the previous methods.

However, although such deep learning models can recover the mixture signals to unmixed vocals and accompaniment tracks brilliantly, these methods are time-consuming and not viable in a resource-limited environment. Another issue is that the context size of the model input is usually longer than a few seconds, which causes a long latency that is not feasible in the real-time scenario.

The main target of this study focuses on extracting the accompaniment track, with lower latency and lower real-time factor (RTF), while remaining the same performance.

1.3 Contributions

The contributions of this study include:

- Accompaniment extraction with lower latency
- Accompaniment extraction with lower RTF

doi:10.6342/NTU202301173

1.4 Structure

There are six chapters in this study:

- Chapter 1. Introduction: Introduce the background, applications, and issues of singing voice separation.
- Chapter 2. Related work: Introduce the previous methods of SVS and their pros and cons.
- Chapter 3. Dataset: Introduce MUSDB18 and the other dataset used in this study.
- Chapter 4. Methods: Elaborate the experimental configuration, metrics, and the methods to improve accuracy and efficiency.
- Chapter 5. Experiments: Display the experimental results and give the discussions.
- Chapter 6. Conclusion and future work: Conclude the results and give the topics to study in the future.

doi:10.6342/NTU202301173

Chapter 2

Related Work

Recently, several methods were proposed to solve the music source separation (MSS) task and its subtask, the singing voice separation (SVS) task. The traditional methods focused on analyzing the spectrogram of the mixture signal by performing matrix operations. However, as a result of the advancement of the hardware, deep-learning-based solutions became feasible and outperformed the previous methods. Currently, most of the solutions for the MSS task are built upon deep neural networks.

These deep learning methods can be roughly divided into two categories, time-domain-based and frequency-domain-based ones. Nonetheless, the hybrid ones were proposed afterward and attained a higher separation effect recently.

doi:10.6342/NTU202301173

2.1 Traditional Approaches

2.1.1 Robust Principal Component Analysis (RCPA)

Huang *et al.*[3] assumed music accompaniment to be in a low-rank subspace and considered singing voices as the relatively sparse part of the music. They solved the RCPA problem to obtain the low-rank matrix L and the sparse matrix S, containing music accompaniment and vocal signals respectively.

2.1.2 Non-negative Matrix Factorization (NMF)

Vembu *et al.*[4] first identified the vocal part and non-vocal part with discriminators such as support vector machines (SVM) and one-layer neural networks. After that, they performed NMF on the non-vocal segments to separate the vocals and accompaniment. However, such a method requires to make an effort to decide the proper number of components.

2.1.3 REpeating Pattern Extraction Technique (REPET)

Rafii *et al.*[5] leveraged the characteristic of the background music, repetition, to separate the voice and music parts. They computed the autocorrelation of the power spectrogram to recognize the periodicity in the signal. Then, the period is used to build a repeating segment model and repeating spectrogram model to acquire the estimated voice and music signals.

doi:10.6342/NTU202301173

2.2 Deep Learning-based Approaches

2.2.1 U-Net

Jansson *et al.*[6] perceived the SVS task as an image-to-image translation. Therefore, they proposed a neural network based on the U-Net architecture, which is a convolutional encoder-decoder architecture initially used in biomedical image segmentation. Since there is a high similarity between the mixture of the music and its background, the Unet-like architecture can leverage its skip connections to reconstruct a high-quality accompaniment track.

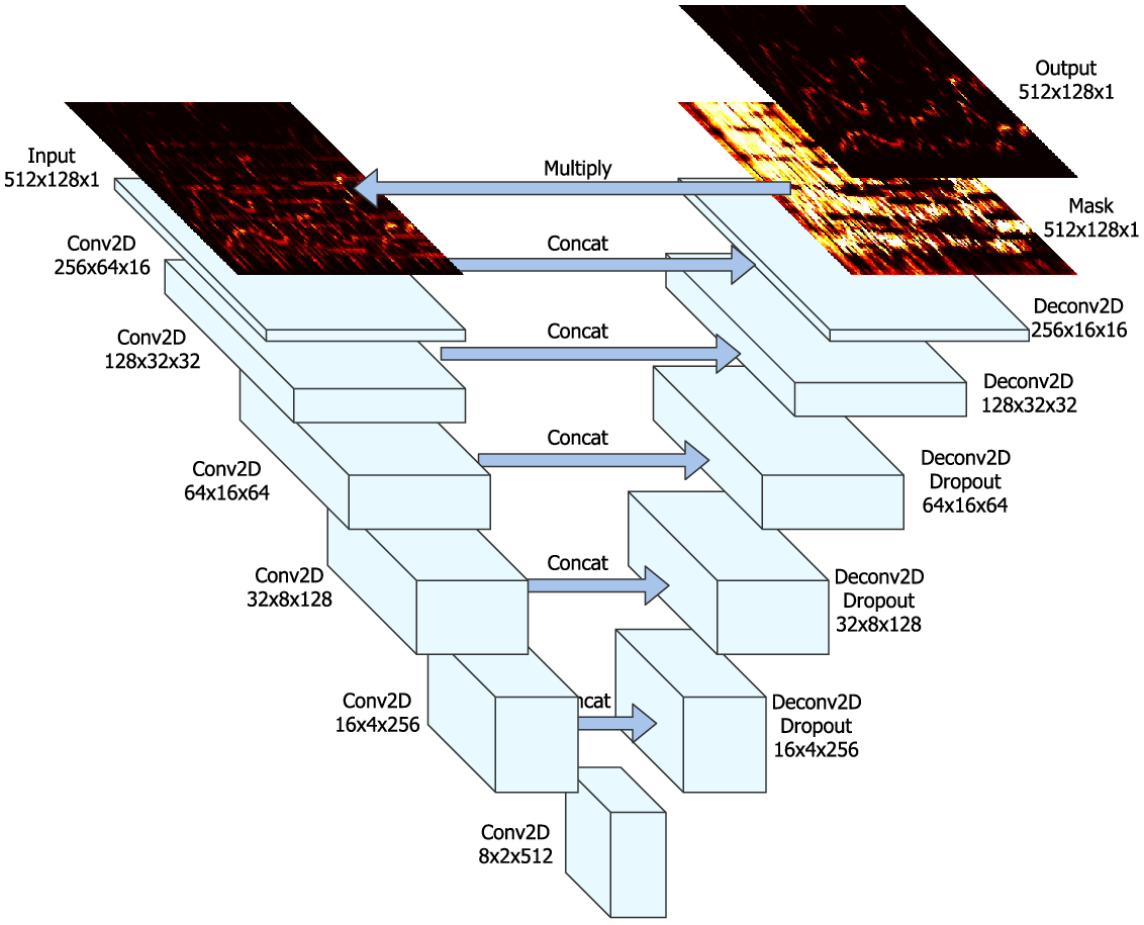


Figure 2.1: U-net architecture [6]

doi:10.6342/NTU202301173

2.2.2 Demucs

Meta[7] also presented their MSS model, Demucs. Based on U-Net architecture and after several improvements, it has been to version 4 now. Demucs was initially focusing on time-domain signal processing to achieve the separation. However, as it evolved to version 3, both time-domain signal and frequency-domain spectrogram are taken into account and won first place in the Music Demixing Challenge 2021[1]. In version 4, by blending the well-known transformer modules into its architecture, Demucs becomes the state-of-the-art approach of the MSS task as this paper is composed.

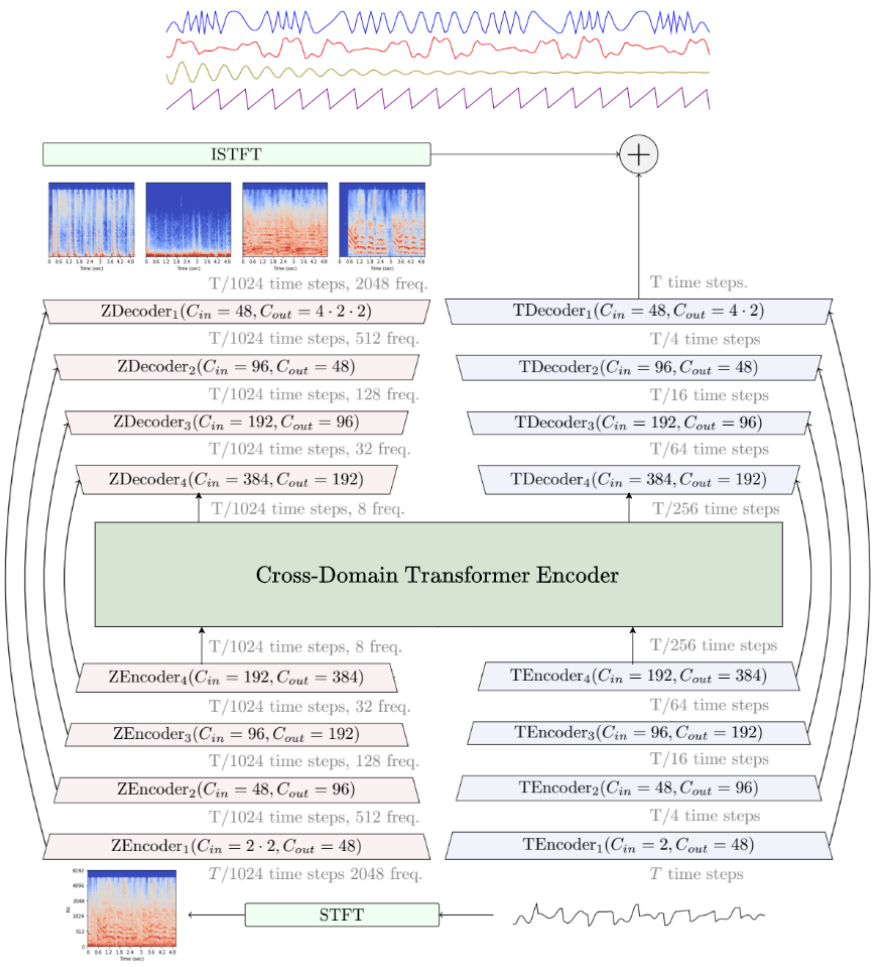


Figure 2.2: Demucs v4 architecture [7]

doi:10.6342/NTU202301173

2.2.3 Multi-Scale Multi-Band DenseNet (MMDenseNet)

Takahashi *et al.*[8] proposed MMDenseNet in 2017, which outperformed the winner of the 2016 Signal Separation Evaluation Campaign (SiSEC)[9]. They extended the DenseNet[10] with up-samplings, block-skipping connections to retrieve both long-term and short-term contextual information and named it MDenseNet.

Furthermore, the band-dedicated MDenseNets are incorporated aiming to analyze high-frequency and low-frequency information individually, since the high-frequency band tends to contain low energies and noise while the low-frequency band may have high energies and tonalities. The outputs of these blocks are concatenated with a full-band MDenseNet and passed into the final dense block to integrate the information.

Due to its low number of parameters, MMDenseNet is an ideal choice when it comes to real-time issues and limited resources. Thus, MMDenseNet is chosen as the baseline model in this study.

doi:10.6342/NTU202301173

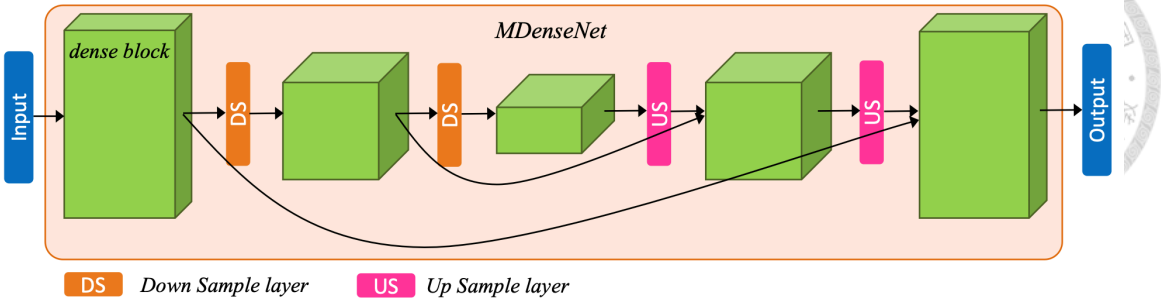


Figure 2.3: MDenseNet architecture [8]

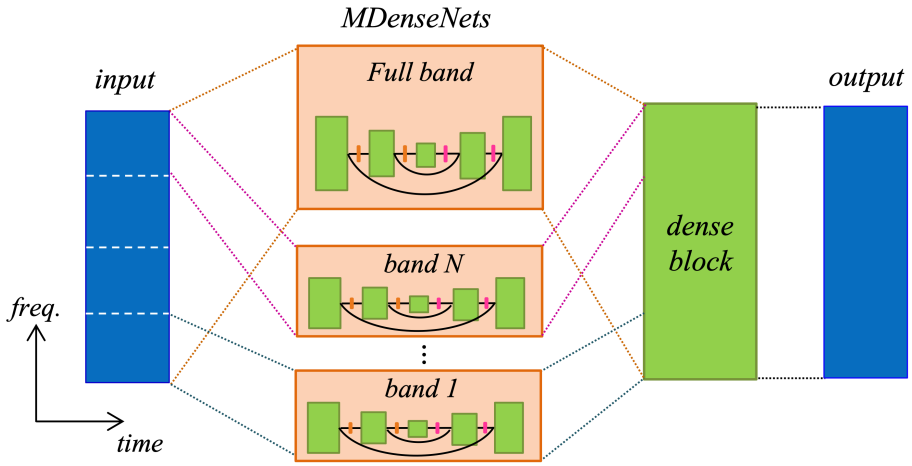


Figure 2.4: MMDenseNet architecture [8]

doi:10.6342/NTU202301173

2.2.4 Real-Time MDenseNet

Huber *et al.*[11] targeted at dealing with real-time, limited computing resources SVS. They extended MDenseNet to facilitate real-time separation issues with techniques such as Mel-scaled mask estimation, deep clustering, parameterized structured pruning, and so on.

The context size can be reduced to 128ms after applying the methods mentioned above. And the decrease in performance is still acceptable.

2.2.5 D3Net

After proposed MMDenseNet[8] and MMDenseLSTM[12], Sony[13] further presented D3Net. It incorporated dilation in DenseNet[10] architecture with multi-dilated convolution layers. Such a method can properly tackle the severe aliasing problem when directly combining DenseNet with dilation. D3Net even achieved the state-of-the-art result at that time.

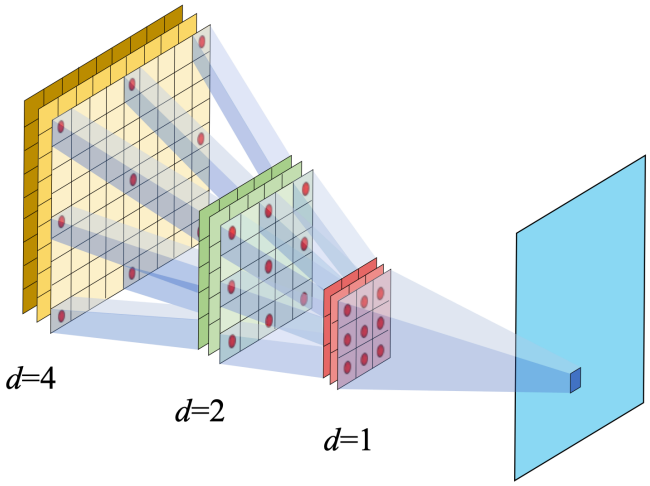


Figure 2.5: Multi-dilated convolution [13]

doi:10.6342/NTU202301173

2.2.6 Training Target

Since there is a high similarity between the SVS task and the speech separation task, which is aiming to separate target speech from background interference. Some different training targets mentioned in [14] can apply to the SVS task as well, such as spectral magnitude mask (SMM)[15] and phase-sensitive mask (PSM)[16].

doi:10.6342/NTU202301173

Chapter 3

Dataset

There are two datasets used in this study: MUSDB18-HQ and some popular songs collected otherwise.

3.1 MUSDB18-HQ[17]

MUSDB18-HQ is a dataset released in the SiSEC 2018 contest. Because of the copyright issue, the open source dataset of the Music source separation (MSS) task is relatively scarce. MUSDB18-HQ is almost the most complete one in this domain.

There are two versions of the dataset, compressed and uncompressed. The compressed version limits the bandwidth of the audio to 16kHz, while the bandwidth of the uncompressed version is up to 22kHz. The high-quality version is used in this study.

Composed of several different sources, there are 100 tracks from DSD 100 dataset[18], 46 tracks from MedleyDB[19], 2 tracks from Native Instruments, and 2 tracks from The Easton Ellises.

doi:10.6342/NTU202301173

The specification of the dataset is as follows.

- Sample rate: 44.1kHz
- Number of channels: stereophonic
- Total Duration: 10 hours
- Training set: 86 songs
- Validation set: 14 songs
- Test set: 50 songs
- Stem: vocals, bass, drums, other

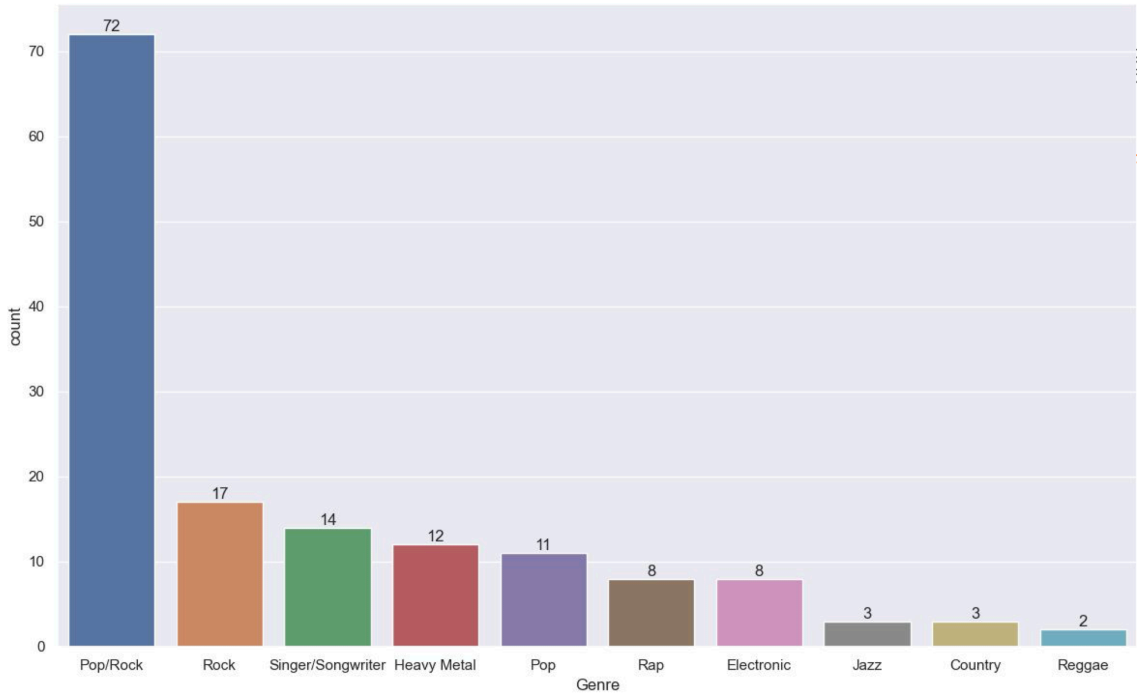


Figure 3.1: MUSDB18-HQ genre distribution

doi:10.6342/NTU202301173

3.2 Popular Music

To be in line with contemporary popular music, other five songs with different genres are chosen as the samples in the subjective tests.

Table 3.1: Popular test samples

Genre	Singer	Name
Hip-hop	O.WEN	官方回答
Jazz	Ella Fitzgerald	All of Me
Pop	周興哲	怎麼了
R&B	Julia Wu	你是不是有點動心
Rock&Roll	蕭敬騰	皮囊

doi:10.6342/NTU202301173

doi:10.6342/NTU202301173

Chapter 4

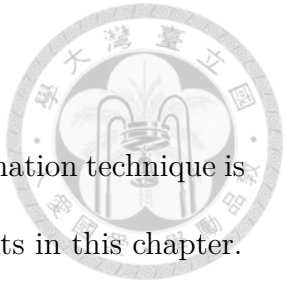
Method

4.1 Problem Definition

In this study, one of the deep learning approaches model, MMDenseNet[8], is chosen as the baseline model. Due to its lightweight property, it is an appropriate choice in the real-time accompaniment extraction task. However, the separation performance decreases rapidly when the input of the model becomes shorter. The shorter input duration also means the estimation of the statistical metrics becomes dubious while applying multichannel Wiener filter[20] as post-processing.

Therefore, performing accompaniment extraction that gets rid of the use of the multichannel Wiener filter, reduces the input context size, and preserves the performance are the main topics to be discussed.

doi:10.6342/NTU202301173



There are three major topics to be discussed in this chapter.

- To surpass the multichannel Wiener filtering, the mask estimation technique is adopted. There are two different masks put into experiments in this chapter.
- To reduce the input context size, several combinations of context size during training and inference processes are discussed in this chapter.
- To preserve the performance, techniques such as multi-dilated convolution are adopted to mitigate the damage caused by the insufficient context size in this chapter.

4.2 Experimental Configuration

4.2.1 Hardware Environment

Due to the confidential issue, the detail of the hardware device is not allowed to be publicized.

4.2.2 Hyperparameters

The hyperparameters of the experiments are listed below.

- Feature extraction
  - Sample rate: upsample to 48kHz
  - Channel: downmix to monophonic
  - Frame size: 2048 samples
  - Hop size: 1024 samples

doi:10.6342/NTU202301173

- Neural network
  - Batch size: 16 samples
  - Optimizer: Adam[21]
  - Loss function: mean squared error (MSE)
  - Learning rate: 1×10^{-3}
  - Learning rate scheduler: stepLR
    - Step size: 5 epochs
    - Multiplicative factor: 0.98
  - Training input duration: 256 frames
  - Inference input duration: 256 frames

4.2.3 Augmentation

The augmentation of the experiments is listed below.

- Apply random gains in the interval [0.25, 1.25]
- Randomly remix the sources from different tracks as in [22]

doi:10.6342/NTU202301173

4.3 Evaluation Metrics

The evaluation metrics used in the following experiments can be categorized into two aspects: accuracy and efficiency.

4.3.1 Accuracy

To evaluate the performance of the estimated accompaniment track, the metric proposed by Vincent *et al.*[23] is used in this study. They defined several metrics, source to distortion ratio (SDR), source to interferences ratio (SIR), source to noises ratio (SNR), and source to artifacts ratio (SAR). Since SDR can roughly represent the effect of separation, the performances of the experiments are evaluated by SDR.

They decomposed the the j-th estimated source $\hat{s_j}$ as

$$\hat{s_j} = s_{target} + e_{interf} + e_{noise} + e_{artif} \tag{4.1}$$

and defined the source-to-distortion ratio (SDR) as

$$SDR := 10\ log_{10}\ \frac{\|s_{target}\|^2}{\|\hat{s_j} - s_{target}\|^2} = 10\ log_{10}\ \frac{\|s_{target}\|^2}{\|e_{interf} + e_{noise} + e_{artif}\|^2} \tag{4.2}$$

where

- s_{target}: the signal of the target source
- e_{interf}: the interferences from other sources
- e_{noise}: the noise in the signal
- e_{artif}: the artifacts from other causes

doi:10.6342/NTU202301173

4.3.2 Efficiency

To evaluate the efficiency, two metrics are used in the following experiments, latency and real-time factor (RTF).

4.3.2.1 Latency

In this study, latency is defined as the time delay that the model needs to wait for enough amount of data to be fed into the network. That is the length of new data needed each time in the block-processing system.

Besides, the latency is also affected by the block size. Reducing the block size can effectively decrease the latency. Therefore, reducing the block size while keeping the performance of separation is another critical topic to be discussed.

4.3.2.2 Real-Time Factor (RTF)

The real-time factor (RTF) is a metric representing the relation between the input duration and the processing time. When the RTF of a system is less than one, it is called a real-time system. The RTF is defined as the following formula.

$$RTF = \frac{processing\_time}{input\_duration} \tag{4.3}$$

doi:10.6342/NTU202301173

4.4 Training target replacement

In this section, several training targets are chosen in the experiments to replace the original one in the MMDenseNet[8], which contains the multi-channel Wiener filter post-processing that is infeasible in the real-time condition.

4.4.1 Mask Estimation

In most of the music source separation (MSS) models, the well-known multi-channel Wiener filter (MWF)[20] post-processing is applied to enhance the effect of separation. To perform the filtering, several terms in the minimum mean squared error estimator need to be estimated. In practice, these terms can be estimated by the expectation-maximization (EM) algorithm[24].

However, such estimation is not feasible in the real-time system. The EM algorithm needs a sufficient amount of consecutive frames to estimate the statistics. With the lack of information, the statistics estimated become unstable and unable to enhance the separation. In addition, the computation resource consumed to execute the algorithm is high, which is not applicable in the real-time condition either.

On the other hand, Huber *et al.*[11] had the network learn a Wiener filter-like mask. The estimated mask would be multiplied by the magnitude spectrogram of the original mixture signal. And the product would be the estimated magnitude spectrogram used to recover the target source signal. With such modification, the system can overpass the time-consuming EM algorithm without severe performance degradation to meet the real-time restriction.

doi:10.6342/NTU202301173

4.4.2 Spectral Magnitude Mask (SMM)

To apply the mask estimation, the spectral magnitude mask (SMM) is one of the instinctive ways to fulfill it. Mentioned in [14] and also called FFT-MASK in [15], the mask is simply defined as the ratio between the magnitude spectrograms of mixture and target source signals.

The estimated magnitude spectrogram can be obtained by multiplying the estimated mask and the magnitude spectrogram of the mixture signal after the short-time Fourier transform (STFT).

One thing to be noted is that the SMM is performed with magnitude only, which means the phase information is merely the duplicate of that of the mixture signal.

The definition of SMM is as follows.

$$SMM(t,f) := \frac{|S(t,f)|}{|Y(t,f)|} \tag{4.4}$$

where

- S: the spectrogram of the target source
- Y: the spectrogram of the mixture signal
- t: the index of the frame
- f: the index of the frequency bin

doi:10.6342/NTU202301173

4.4.3 Phase-Sensitive Mask (PSM)

To make the most of phase information in the spectrogram, [16] proposed a phase-sensitive objective function named phase-sensitive mask (PSM). The only difference between PSM and SMM is the cosine function, which can be regarded as a projection from the mixture to the target source.

Therefore, the substitution of PSM can take both magnitude and phase information into account without much additional cost.

The definition of PSM is as follows.

$$PSM(t,f) := \frac{|S(t,f)|}{|Y(t,f)|} cos\theta \tag{4.5}$$

where

- S: the spectrogram of the target source
- Y: the spectrogram of the mixture signal
- t: the index of the frame
- f: the index of the frequency bin
- θ: the difference between the mixture phase and the target source phase

doi:10.6342/NTU202301173

4.5 Accuracy Improvement

To further improve the performance of separation, multi-dilated convolution[13] is incorporated into the MMDenseNet. Apart from multi-dilated convolution, some other adjustments are applied to the architecture of the model to increase the model complexity.

4.5.1 Multi-Dilated Convolution

In [13], Sony proposed the multi-dilated convolution for DenseNet, which discussed the aliasing problem when directly replacing the convolution operation with the dilated convolution in the DenseNet. Therefore, they proposed a solution named multi-dilated convolution to deal with the problem.

After such modification, information can be modeled with different resolutions simultaneously in a single layer without the increasing of computing resources.

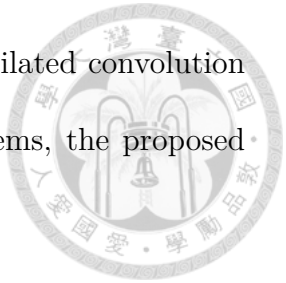
Denote the output of the l-th convolution layer within a dense block as

$$x_l = \psi([x_0, x_1, ..., x_{l-1}]) \circledast k_l \tag{4.6}$$

where

- ψ: the composition operation of batch normalization and nonlinearity
- $[x_0, x_1, ..., x_{l-1}]$: the concatenation of feature maps from 1 to l-th layers
- $\circledast$: the convolution operation

doi:10.6342/NTU202301173



Since the naive incorporation of replacing convolution with dilated convolution may create several blind spots leading to several aliasing problems, the proposed multi-dilated convolution is to overcome the problem.

Denote the multi-dilated convolution $\circledast_l^m$ as

$$Y_l \circledast_l^m k_l = \sum_{i=0}^{l-1} y_i \circledast_{d_i} k_l^i \tag{4.7}$$

where

- $Y_l = [y_0, y_1, ..., y_l] = \psi([x_0, x_1, ..., x_l])$: the composite layer output
- k_l^i: the subset of filters corresponding to the i-th skip connection
- $d_i = 2^i$: the dilation factor
- $\circledast_d$: the dilated convolution

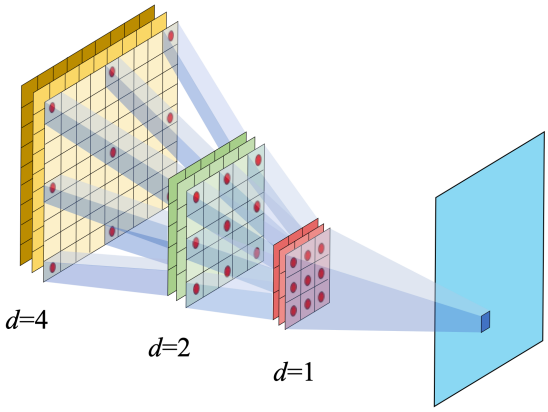
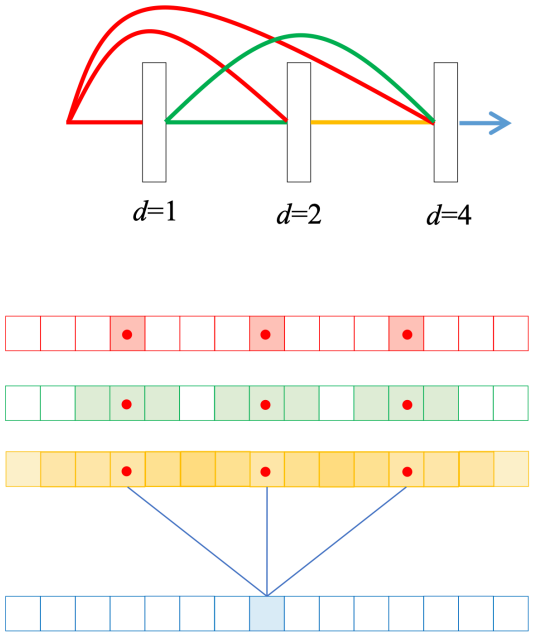
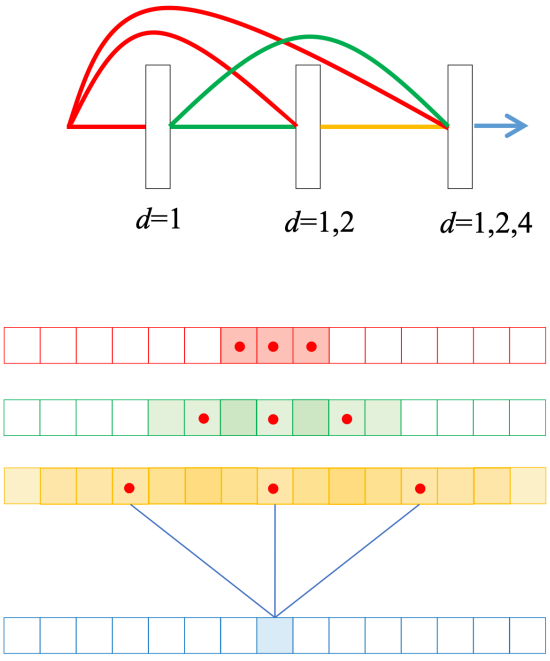


Figure 4.1: Multi-dilated convolution [13]

doi:10.6342/NTU202301173



(a) Naive dilated convolution



(b) Multi-dilated convolution

Figure 4.2: Comparison of receptive fields of different dilated convolutions [13]

doi:10.6342/NTU202301173

4.5.2 Double Number of Channels

In several experiments, increasing the complexity of the neural network (NN) model may improve the accuracy. Thus, in the following experiment, the complexity of the model is increased by doubling the growth rates of the dense blocks to enhance the performance.

The variables of the following tables are denoted as:

- t: the frame index
- f: the frequency bin index
- ch: the number of channels
- k: the growth rate
- L: the number of layers
- t.conf: transposed convolution

doi:10.6342/NTU202301173

Table 4.1: The original MMDenseNet architecture

Layer	low	high	full	scale
band split	first half	last half	-	1
conv (t, f, ch)	3, 4, 32	3, 3, 32	3, 4, 32	
dense 1 (k, L)	14, 4	10, 3	6, 2	
down sample	pool	pool	pool	$\frac{1}{2}$
dense 2 (k, L)	16, 4	10, 3	6, 2	
down sample	pool	pool	pool	$\frac{1}{4}$
dense 3 (k, L)	16, 4	10, 3	6, 2	
down sample	pool	pool	pool	$\frac{1}{8}$
dense 4 (k, L)	16, 4	10, 3	6, 4	
up sample	t.conv	t.conv	t.conv	$\frac{1}{4}$
concat.	low dense 3	high dense 3	full dense 3	
dense 5 (k, L)	16, 4	10, 3	6, 2	
up sample	t.conv	t.conv	t.conv	$\frac{1}{2}$
concat.	low dense 2	high dense 2	full dense 2	
dense 6 (k, L)	16, 4	10, 3	6, 2	
up sample	t.conv	t.conv	t.conv	1
concat.	low dense 1	high dense 1	full dense 1	
dense 7 (k, L)	16, 4	10, 3	6, 2	
concat. (axis)	frequency		-	1
concat. (axis)	channel			
dense 8 (k, L)	4, 2			
conv (t, f, ch)	1, 2, 2			

doi:10.6342/NTU202301173

Table 4.2: The modified MMDenseNet architecture

Layer	low	high	full	scale
band split	first half	last half	-	1
conv (t, f, ch)	3, 4, 64	3, 3, 64	3, 4, 64	
dense 1 (k, L)	28, 4	20, 3	12, 2	
down sample	pool	pool	pool	$\frac{1}{2}$
dense 2 (k, L)	32, 4	20, 3	12, 2	
down sample	pool	pool	pool	$\frac{1}{4}$
dense 3 (k, L)	32, 4	20, 3	12, 2	
down sample	pool	pool	pool	$\frac{1}{8}$
dense 4 (k, L)	32, 4	20, 3	12, 4	
up sample	t.conv	t.conv	t.conv	$\frac{1}{4}$
concat.	low dense 3	high dense 3	full dense 3	
dense 5 (k, L)	32, 4	20, 3	12, 2	
up sample	t.conv	t.conv	t.conv	$\frac{1}{2}$
concat.	low dense 2	high dense 2	full dense 2	
dense 6 (k, L)	32, 4	20, 3	12, 2	
up sample	t.conv	t.conv	t.conv	1
concat.	low dense 1	high dense 1	full dense 1	
dense 7 (k, L)	32, 4	20, 3	12, 2	
concat. (axis)	frequency		-	1
concat. (axis)	channel			
dense 8 (k, L)	8, 2			
conv (t, f, ch)	1, 2, 2			

doi:10.6342/NTU202301173

4.6 Efficiency Improvement

To achieve the real-time requirement, the relation of input duration in training and inference processes is worth to be discussed in this section. Apart from the input duration, context aggregation is another effective method to reduce the latency.

4.6.1 Shorter Input Duration

To improve efficiency, the first idea is to directly reduce the number of frames during the inference process. However, this may lead to inconsistency between the duration of training and inference processes, especially in the short input duration. Therefore, the relation between the lengths of training and inference processes is one of the topics of concern.

doi:10.6342/NTU202301173

4.6.2 Context Aggregation

Another technique to reduce the latency is to aggregate the context, which is mentioned in [11]. That is, store the information in the past and concatenate the current information.

With such a method, the relation between input duration, real-time factor (RTF), and latency can be formulated as the equation below.

$$latency = \lceil input\_duration \times RTF \rceil \tag{4.8}$$

From the equation, input duration and RTF are two key factors that dominate the latency. The shorter input duration and lower RTF lead to lower latency.

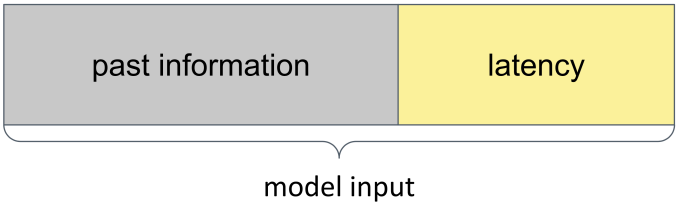


Figure 4.3: Context aggregation

doi:10.6342/NTU202301173

Chapter 5

Result

In this chapter, the experimental results of the methods in chapter 4 would be discussed. There are three categories of the experiments, training target, accuracy improvement, and efficiency improvement.

The baseline model of the following experiments is the original MMDenseNet[8] with multichannel Wiener filter[20] post-processing, trained with the same configuration as 4.2.2. And the currently state-of-the-art model HT Demucs[7] would be compared in the end as well. The latencies of models in this chapter are estimated by their input context sizes.

doi:10.6342/NTU202301173

5.1 Training Target

In this section, the experimental results of applying the spectral magnitude mask (SMM)[15] and the phase-sensitive mask (PSM)[16] would be discussed.

5.1.1 Spectral Magnitude Mask (SMM)

After replacing the original training target (i.e. the magnitude spectrogram) with SMM, the computation of the multichannel Wiener filter is passed over. The computational cost can be significantly decreased with slight performance degradation accordingly.

Table 5.1: Comparison of models with SMM

Model	No. of Param.	Latency (second)	RTF	SDR
Baseline	0.33M	5.46	Time Limit Exceeded	13.87
SMM			0.36	13.53

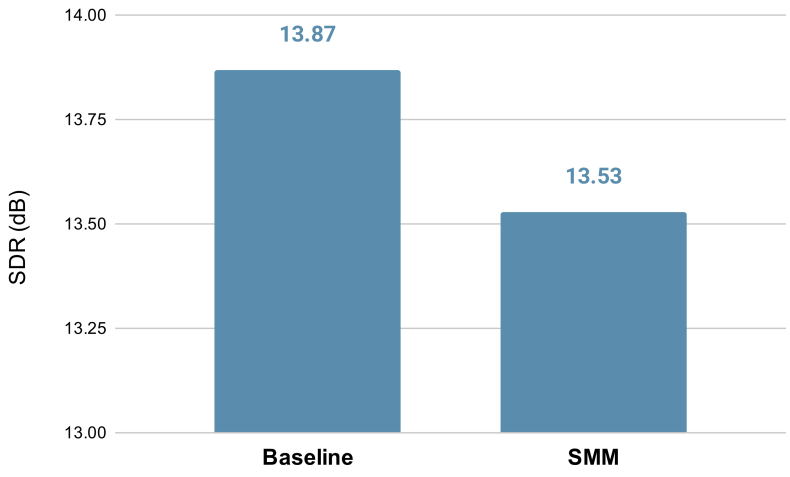


Figure 5.1: Comparison of models with SMM

doi:10.6342/NTU202301173

5.1.2 Phase-Sensitive Mask (PSM)

To make use of the phase information, the PSM is further substituted for the SMM. With the same computational complexity, such substitution improves the performance of separation, even higher than that of the baseline model.

Table 5.2: Comparison of models with different training targets

Model	No. of Param.	Latency (second)	RTF	SDR
Baseline	0.33M	5.46	Time Limit Exceeded	13.87
SMM			0.36	13.53
PSM				13.93

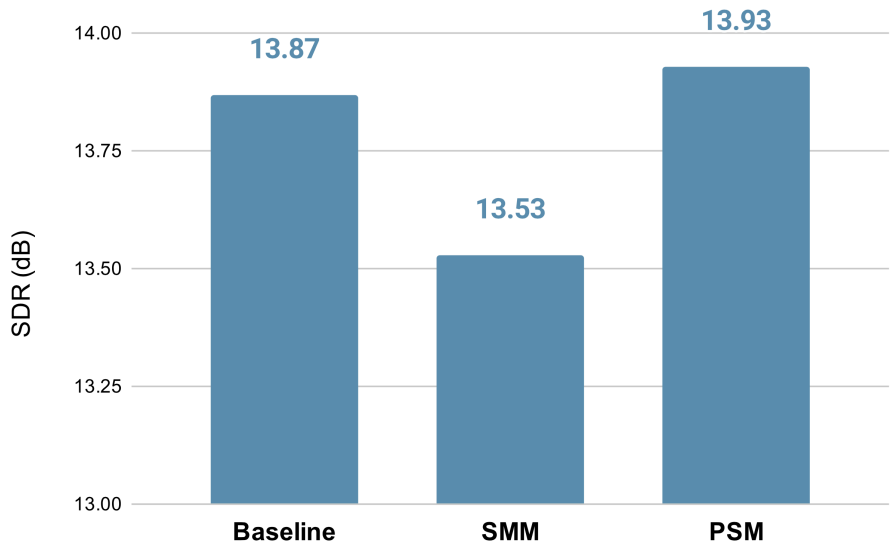


Figure 5.2: Comparison of models with different training targets

doi:10.6342/NTU202301173

5.2 Accuracy Improvement

In this section, the improvement of accuracy is achieved by incorporating multi-dilated convolution[13] and increasing the model complexity. The degree of improvement would be discussed.

5.2.1 Multi-dilated convolution

After incorporating multi-dilated convolution into the original MMDenseNet architecture, the performance of separation becomes higher without increasing the number of parameters and the real-time factor (RTF). This may be due to the characteristic of multi-dilated convolution that can retrieve information with different resolutions simultaneously. And there is no need to increase the number of parameters.

Table 5.3: Layer depth and dilation rate within a dense block

Layer depth	Dilation rate
1	1
2	2
3	4
4	8

Table 5.4: Comparison of models with multi-dilated convolution

Mask	Multi-dilated Conv.	No. of Param.	Latency (second)	RTF	SDR
SMM	x	0.33M	5.46	0.36	13.53
PSM					13.93
SMM	o				13.99
PSM					14.25

doi:10.6342/NTU202301173

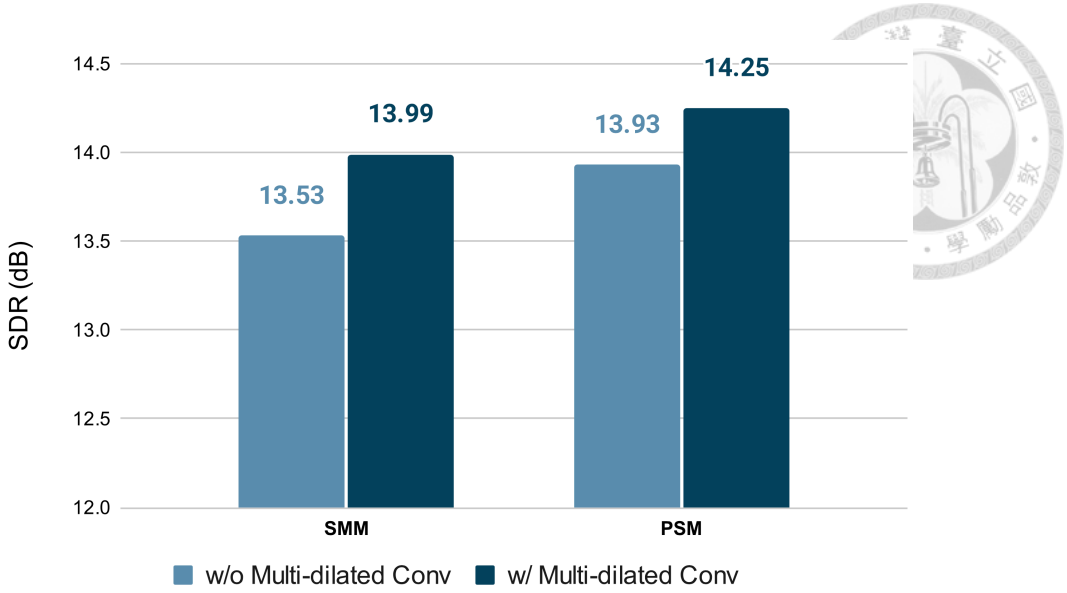


Figure 5.3: Comparison of models with multi-dilated convolution

5.2.2 Double Number of Channels

The model complexity is increased by doubling the number of channels in the convolution layers. The performance of separation indeed becomes higher. However, the RTF becomes higher as well, which is a trade-off between performance and efficiency.

The number of parameters is increased from 0.33M to 1.33M.

Table 5.5: Comparison of models with double channels

Mask	Multi-dilated Conv.	2x channel	No. of Param.	Latency (s)	RTF	SDR
SMM	o	x	0.33M	5.46	0.36	13.99
PSM						14.25
SMM		o	1.33M		0.70	14.22
PSM						14.64

doi:10.6342/NTU202301173

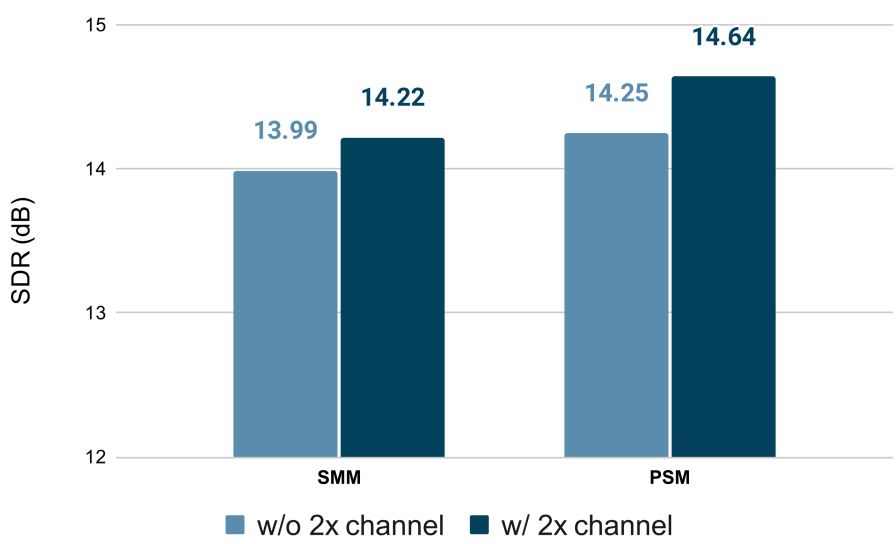


Figure 5.4: Comparison of models with double channels

5.3 Efficiency Improvement

To improve the efficiency, the relation of input duration between the training and inference process and the context aggregation technique would be discussed in this section.

The models in the following experiments use the PSM as the training target, incorporating multi-dilated convolution, but without double the number of channels. The experiments with other configurations are presented in appendix 1.

doi:10.6342/NTU202301173

5.3.1 Shorter input duration

5.3.1.1 SDR

To reduce the latency, the straightforward method is to reduce the input duration in the inference process as well. However, such a method leads to considerable performance degradation. Therefore, the experiment indicates that the performance becomes higher when the input duration is matched in the training and inference processes, especially with the shorter input duration.

Table 5.6: Comparison of models with different input duration (SDR). The value colored red is the highest score within each column.

		Inference duration (frame)					
		16	24	32	64	128	256
Training duration (frame)	16	13.02	13.03	13.08	13.23	13.28	13.36
	24	12.66	13.40	13.37	13.44	13.26	13.28
	32	12.52	13.03	13.44	13.41	13.51	13.53
	64	12.75	13.08	13.30	13.82	14.05	14.09
	128	12.07	12.78	13.12	13.63	13.93	13.89
	256	11.64	12.21	12.81	13.68	14.04	14.25

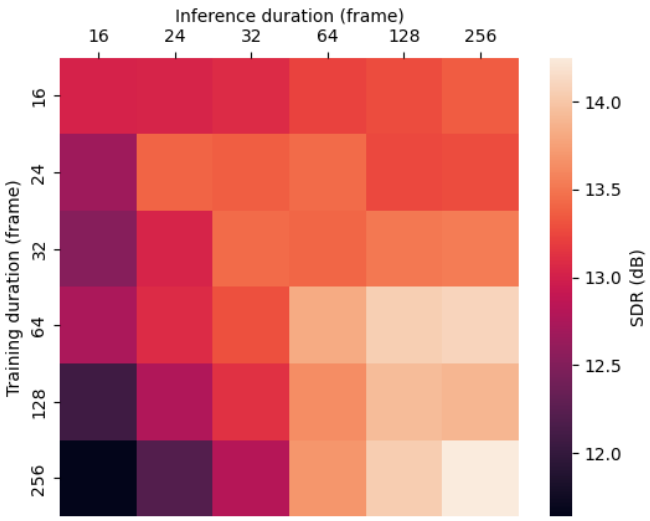


Figure 5.5: Comparison of models with different input duration (SDR).

doi:10.6342/NTU202301173

5.3.1.2 RTF

The RTF, in the other aspect, is not affected by the input duration apparently. The lower points may be due to the hardware design to accelerate the efficiency in certain circumstances such as single instruction, multiple data (SIMD).

Table 5.7: Comparison of models with different input duration (RTF).

Duration (frame)	Duration (second)	Runtime (second)	RTF
16	0.341	0.125	0.366
24	0.512	0.188	0.367
32	0.683	0.236	0.346
64	1.365	0.517	0.379
128	2.731	1.001	0.367
256	5.461	1.943	0.356

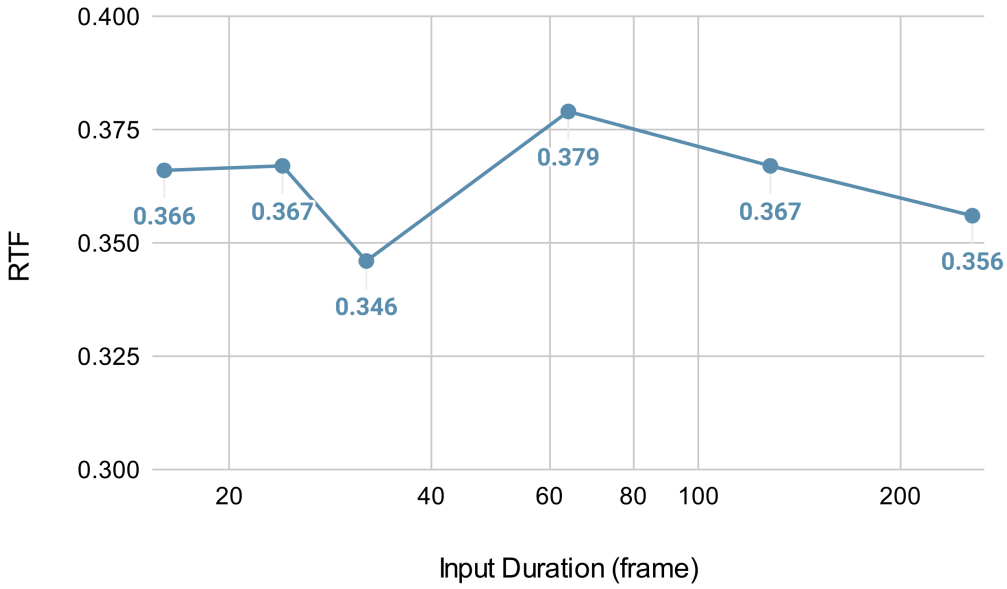


Figure 5.6: Comparison of models with different input duration (RTF).

doi:10.6342/NTU202301173

5.3.2 Context Aggregation

To further lower the latency, context aggregation seems to be an appropriate solution. The experiment indicates that the latency can be decreased to about 500ms while remaining the same performance as the baseline model. The latencies in this section are estimated by the input duration and the RTF in the previous section.

Table 5.8: Comparison of models with different context aggregations.

Duration (frame)	RTF	Latency (frame)	Latency (second)
16	0.366	6	0.128
24	0.367	9	0.192
32	0.346	12	0.256
64	0.379	25	0.533
128	0.367	47	1.003
256	0.356	92	1.963

To cater to various applications, several candidates with different latencies are listed below.

Table 5.9: Comparison of models.

Model	Latency (second)	SDR (dB)
baseline	5.460	13.87
HT Demucs	3.120	16.64
PSM (multi-dilated conv.)	0.128	13.02
PSM (multi-dilated conv.)	0.192	13.40
SMM (multi-dilated conv., 2x channel)	0.491	13.63
PSM (multi-dilated conv., 2x channel)	0.960	14.13

doi:10.6342/NTU202301173

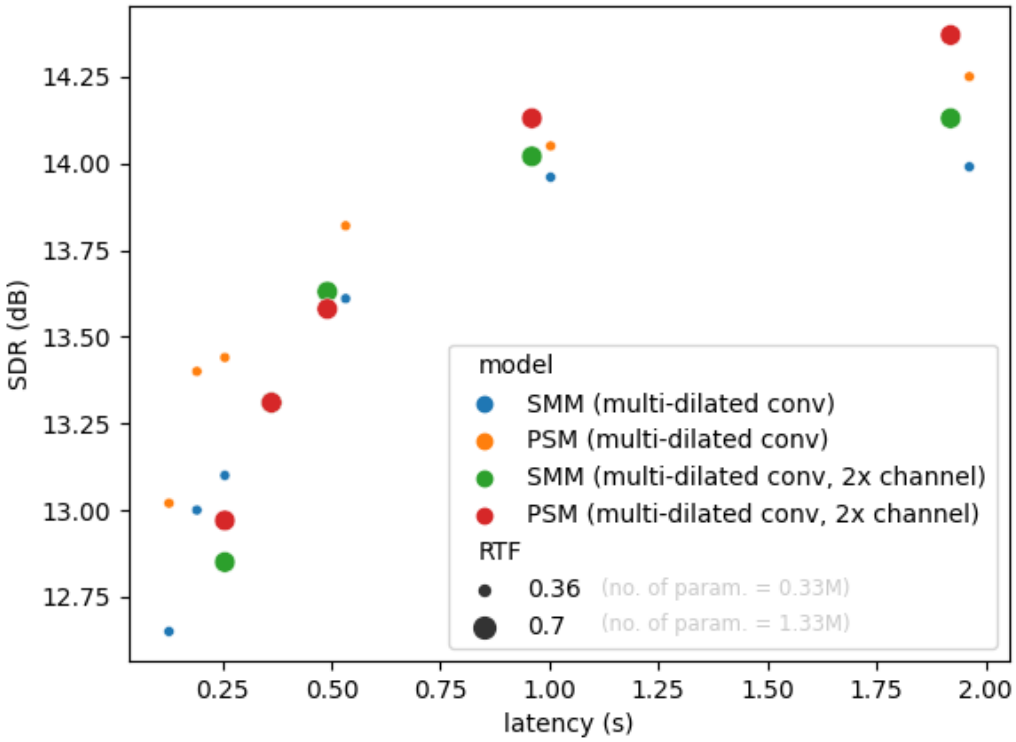


Figure 5.7: Comparison of models.

doi:10.6342/NTU202301173

Chapter 6

Conclusion and Future Work

6.1 Conclusion

From the experiments in chapter 5, there are several main points listed as follows.

1. The experiment in 5.1.1 indicates that the mask estimation is one of the solutions to overpass the time-consuming computation of multichannel Wiener filter post-processing when it comes to the real-time issue.

2. The experiments in 5.1.2 and 5.2.1 indicate that the techniques such as phase-sensitive mask (PSM)[16] and multi-dilated convolution can enhance the performance of separation without additional computational resources.

3. The experiment in 5.2.2 indicates that increasing the model complexity can further uplift the performance while raising the computing power as well.

4. The experiment in 5.3.1 indicates that the performance decreases rapidly as the input duration goes down, and the SDR becomes higher when the duration of training and inference match.

doi:10.6342/NTU202301173

5. The experiment in 5.3.2 indicates that the latency can be reduced effectively to meet the real-time applications with context aggregation.

6.2 Future Work

There are still other techniques that can be incorporated to deal with real-time singing voice separation (SVS). Some of the possible experiments are listed below.

1. Although the baseline model in this study is MMDenseNet[8], it is possible to adopt methods in 4 to other model such as MMDenseLSTM[12] and D3Net[13] to reach a higher performance. Also, the incorporation of attention[25] units is worthwhile to be experimented with.

2. Since some of the methods to enhance the accuracy in this study are focused on replacing with different training targets. The augmentation of pitch-aware remixing[26] may further enhance the performance as well.

3. The finetuning process with semi-supervised data sampling proposed in [27] is another method to deal with the limited amount of labeled data. The performance may be further improved.

4. Apart from SMM and PSM, the complex ideal ratio mask (cIRM)[28] is another possible choice of training target. Different from SMM and PSM, the cIRM can perfectly recover the signal. It is possible to raise the performance as well.

doi:10.6342/NTU202301173

5. Though the modification of the model architecture only takes accounted for a small part of this study, it is worth performing the experiments to adjust the number of bands in the MMDenseNet, the depth of each dense block, and other scaling factors to increase the model complexity.

doi:10.6342/NTU202301173

doi:10.6342/NTU202301173

Bibliography

[1] Y. Mitsufuji, G. Fabbro, S. Uhlich, and F.-R. Stöter, *Music demixing challenge 2021*, 2021. arXiv: 2108.13559 [eess.AS].

[2] F. Li and M. Akagi, "Blind monaural singing voice separation using rank-1 constraint robust principal component analysis and vocal activity detection," *Neurocomputing*, vol. 350, pp. 44–52, 2019.

[3] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 57–60. DOI: 10.1109/ICASSP.2012.6287816.

[4] S. Vembu and S. Baumann, "Separation of vocals from polyphonic audio recordings .," Jan. 2005, pp. 337–344.

[5] Z. Rafii and B. Pardo, "Repeating pattern extraction technique (repet): A simple method for music/voice separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 1, pp. 73–84, 2013. DOI: 10.1109/TASL.2012.2213249.

[6] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep u-net convolutional networks," 2017.

[7] S. Rouard, F. Massa, and A. Défossez, *Hybrid transformers for music source separation*, 2022. arXiv: 2211.08553 [eess.AS].

[8] N. Takahashi and Y. Mitsufuji, "Multi-scale multi-band densenets for audio source separation," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, IEEE, 2017, pp. 21–25.

[9] A. Liutkus, F.-R. Stöter, Z. Rafii, *et al.*, "The 2016 signal separation evaluation campaign," in *Latent Variable Analysis and Signal Separation: 13th International Conference, LVA/ICA 2017, Grenoble, France*, Springer International Publishing, 2017, pp. 323–332.

[10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

doi:10.6342/NTU202301173

[11] M. Huber, G. Schindler, C. Schörkhuber, W. Roth, F. Pernkopf, and H. Fröning, "Towards real-time single-channel singing-voice separation with pruned multi-scaled densenets," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 806–810. DOI: 10.1109/ICASSP40776.2020.9053542.

[12] N. Takahashi, N. Goswami, and Y. Mitsufuji, "Mmdenselstm: An efficient combination of convolutional and recurrent neural networks for audio source separation," in *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2018, pp. 106–110. DOI: 10.1109/IWAENC.2018.8521383.

[13] N. Takahashi and Y. Mitsufuji, "D3net: Densely connected multidilated densenet for music source separation," *arXiv preprint arXiv:2010.01733*, 2020.

[14] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018. DOI: 10.1109/TASLP.2018.2842159.

[15] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1849–1858, 2014. DOI: 10.1109/TASLP.2014.2352935.

[16] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 708–712. DOI: 10.1109/ICASSP.2015.7178061.

[17] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, *Musdb18-hq - an uncompressed version of musdb18*, Aug. 2019. DOI: 10.5281/zenodo.3338373. [Online]. Available: https://doi.org/10.5281/zenodo.3338373.

[18] A. Liutkus, F.-R. Stöter, Z. Rafii, *et al.*, "The 2016 signal separation evaluation campaign," in *Latent Variable Analysis and Signal Separation - 12th International Conference, LVA/ICA 2015, Liberec, Czech Republic, August 25-28, 2015, Proceedings*, P. Tichavský, M. Babaie-Zadeh, O. J. Michel, and N. Thirion-Moreau, Eds., Cham: Springer International Publishing, 2017, pp. 323–332.

[19] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "Medleydb: A multitrack dataset for annotation-intensive mir research.," in *ISMIR*, vol. 14, 2014, pp. 155–160.

[20] A. Liutkus and F.-R. Stöter, *Norbert: Multichanne-wiener filtering*, version v0.2.1, Sep. 2019. DOI: 10.5281/zenodo.3386463. [Online]. Available: https://doi.org/10.5281/zenodo.3386463.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

doi:10.6342/NTU202301173

[22] S. Uhlich, M. Porcu, F. Giron, *et al.*, "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)*, 2017, pp. 261–265. DOI: 10.1109/ICASSP.2017.7952158.

[23] 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, 2006. DOI: 10.1109/TSA.2005.858005.

[24] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society: series B (methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[25] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[26] S. Yuan, Z. Wang, U. Isik, *et al.*, "Improved singing voice separation with chromagram-based pitch-aware remixing," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 111–115. DOI: 10.1109/ICASSP43922.2022.9747612.

[27] Y. Luo and J. Yu, "Music source separation with band-split rnn," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1893–1901, 2023. DOI: 10.1109/TASLP.2023.3271145.

[28] D. S. Williamson, Y. Wang, and D. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 3, pp. 483–492, 2016. DOI: 10.1109/TASLP.2015.2512042.

doi:10.6342/NTU202301173

doi:10.6342/NTU202301173

Appendix A

Models with different input duration

The experiment of logarithm feature compression is conducted as well, which means the feature is first converted to the logarithm domain before feeding into the model.

The original input x is converted to the logarithm domain feature y by the formula below.

$$y = log_e(1 + x) \tag{A.1}$$

doi:10.6342/NTU202301173

A.1 SMM w/ Multi-Dilated Convolution

The real-time factor (RTF) with this configuration is about 0.36.

Table A.1: Comparison of models with different input duration using SMM with multi-dilated convolution (SDR).

		Inference duration (frame)					
		16	24	32	64	128	256
Training duration (frame)	16	12.65	12.62	12.52	12.41	12.55	12.74
	24	12.49	13.00	13.12	12.92	13.18	13.21
	32	12.40	12.85	12.94	13.09	13.34	13.36
	64	13.30	12.84	13.10	13.61	13.72	13.75
	128	12.06	12.62	12.85	13.55	13.89	13.74
	256	12.00	12.54	12.99	13.48	13.96	13.99

A.2 LogSMM w/ Multi-Dilated Convolution

The real-time factor (RTF) with this configuration is about 0.36.

Table A.2: Comparison of models with different input duration using SMM with multi-dilated convolution and logarithm feature compression (SDR).

		Inference duration (frame)					
		16	24	32	64	128	256
Training duration (frame)	16	12.50	12.91	12.89	12.87	12.94	12.98
	24	12.65	13.14	13.24	12.84	12.85	12.81
	32	12.29	13.02	13.29	13.25	13.22	13.23
	64	12.42	12.95	13.26	13.73	13.75	13.71
	128	11.91	12.48	12.93	13.65	13.91	13.78
	256	12.07	12.67	12.87	12.70	13.87	13.93

doi:10.6342/NTU202301173

A.3 SMM w/ Multi-Dilated Convolution, 2x Channel

The real-time factor (RTF) with this configuration is about 0.70.

Table A.3: Comparison of models with different input duration using SMM with multi-dilated convolution and double number of channels (SDR).

		Inference duration (frame)					
		16	24	32	64	128	256
Training duration (frame)	16	12.50	12.65	12.85	12.98	12.76	13.02
	24	12.85	13.31	13.22	13.40	13.38	13.38
	32	12.67	13.21	13.37	13.36	13.39	13.47
	64	12.17	12.98	13.31	13.90	13.88	13.83
	128	12.39	13.21	13.38	14.02	14.10	14.08
	256	12.10	12.77	13.11	13.85	14.13	14.22

A.4 LogSMM w/ Multi-Dilated Convolution, 2x Channel

The real-time factor (RTF) with this configuration is about 0.70.

Table A.4: Comparison of models with different input duration using SMM with multi-dilated convolution, double number of channels, and logarithm feature compression (SDR).

		Inference duration (frame)					
		16	24	32	64	128	256
Training duration (frame)	16	13.05	13.29	13.31	13.27	13.40	13.32
	24	12.81	13.31	13.31	13.36	13.43	13.43
	32	12.66	13.22	13.53	13.19	13.50	13.62
	64	12.09	13.23	13.63	13.88	13.80	13.98
	128	11.28	11.87	12.41	13.05	13.83	13.76
	256	11.77	12.57	13.04	13.74	14.10	14.31

doi:10.6342/NTU202301173

A.5 PSM w/ Multi-Dilated Convolution, 2x Channel

The real-time factor (RTF) with this configuration is about 0.70.

Table A.5: Comparison of models with different input duration using PSM with multi-dilated convolution and double number of channels (SDR).

		Inference duration (frame)					
Training duration (frame)		16	24	32	64	128	256
	16	12.70	13.11	13.07	13.04	13.13	13.23
	24	12.97	13.31	13.30	13.12	13.31	13.42
	32	13.37	13.00	13.14	13.42	13.63	13.59
	64	12.06	12.80	13.20	13.60	13.77	13.75
	128	12.56	13.19	13.45	14.05	14.26	14.37
	256	12.44	12.99	13.58	14.13	14.13	14.22

doi:10.6342/NTU202301173