Assistive Technology Using Machine Learning Based on Multi-Domain Data for Articulation Disorders

Takashima, Yuki

(Degree)
博士（工学）

(Date of Degree)
2020-03-25

(Date of Publication)
2021-03-01

(Resource Type)
doctoral thesis

(Report Number)
甲第7781号

(URL)
https://hdl.handle.net/20.500.14094/D1007781

Doctoral Thesis

Assistive Technology Using Machine Learning Based on Multi-Domain Data for Articulation Disorders

構音障害者のための 複数ドメインのデータに基づく 機械学習を用いた支援技術

Yuki TAKASHIMA

高島 悠樹

Graduate School of System Informatics

Kobe University

A thesis submitted for PhD. degree

Jan. 2020

Doctoral Thesis

Assistive Technology Using Machine Learning Based on Multi-Domain Data for Articulation Disorders

Yuki TAKASHIMA

The person with an articulation disorder has problems forming speech sounds properly. Although the voice is one of the most natural communication tools for human beings, the voices of persons with articulation disorders are difficult to understand for people around them. To improve the quality of their life, there is a critical need for voice-driven assistive systems.

In the field of speech processing, numerous researches have been proposed and remarkable progress has been made in fields with access to a large amount of training data. However, there has been very little benefit for persons with speech disorders because these systems are trained on "typical" speech. To achieve a system that works well for disordered speech, in this paper, we tackle the following two problems: limited data availability and difference in a speech style. First, it is difficult to collect a sufficient amount of speech data to train the model owing to a physical burden. Second, their speech style differs significantly from that of physically-unimpaired persons. Therefore, an approach specifically tailored to overcome these problems of impaired speech data is needed.

To solve these problems, this paper proposes the following two approaches: conversion and recognition. Voice conversion (VC) is a technique for converting paralinguistic information in speech, while preserving the linguistic information in the utterance. Although the most popular VC application is speaker conversion, VC also can be applied for emotion conversion and assistive technology.

Automatic speech recognition (ASR) is a technique for translating human speech into text by a computer program.

VC can be applied to improve the intelligibility of disordered speech. However, the conventional VC methods such as the Gaussian mixture model-based one require parallel data to train the models. Parallel data is aligned speech data from the source and the target speakers so that each frame of the source speaker's data corresponds to that of the target speaker. Because both speakers must utter the same articles, the cost to collect such data is high. We propose a dictionary learning framework using non-negative Tucker decomposition where the dictionary denotes the weight matrix to be used for conversion. This method allows us to use naturally spoken speech data without any constraints of the data structure to train the model.

To train the ASR model using a sufficient amount of training data, we propose a transfer learning method using multiple databases. To be specific, we use three different sources of data: speech data from a target speaker with an articulation disorder, speech data from physically unimpaired persons in the same target language, and speech data of persons with articulation disorders in other languages. We assume that the phonetic or linguistic characteristics are dependent on language and the dysarthric characteristics are independent of language. Our method can transfer these characteristics into the target model of a person with an articulation disorder from the additional databases.

The face or lip movements can be complementary information for ASR. In the case of persons with articulation disorders resulting from severe hearing loss, their speech style is quite different from those of people without hearing loss. Some people with hearing loss can communicate by reading the lip shape and making the proper lip shape. Therefore, we propose an audio-visual speech recognition system for them. In this study, we extract a bottleneck feature that represents aggregated phonetic information from neural networks.

As mentioned above, we raise two problems and introduce two approaches. Our proposed methods contribute to not only our evaluated tasks but also various assistive technologies in the fields where a large amount of data may not be available.

Contents

List of Figures

List of Tables

Declaration

I herewith declare that I have produced this paper without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such. This paper has not previously been presented in identical or similar form to any other german or foreign examination board.

The related contents in this thesis have been previously published or submitted for publication by the author. A complete list of publications can be found on *pp.* xv-xix.

Publication List

Journal Papers

1. Yuki Takashima, Yasuhiro Kakihara, Ryo Aihara, Tetsuya Takiguchi, Yasuo Ariki, Nobuyuki Mitani, Kiyohiro Omori, Kaoru Nakazono: "Audio-Visual Speech Recognition Using Convolutive Bottleneck Networks for a Person with Severe Hearing Loss," *IPSJ Transactions on Computer Vision and Applications*, Vol. 7, pp. 64-68, 2015, doi: 10.2197/ipsjtcva.7.64.

2. Yuki Takashima, Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki: "Non-parallel dictionary learning for voice conversion using non-negative Tucker decomposition," *EURASIP Journal on Audio, Speech, and Music Processing*, 11 pages, 2019:17 doi: 10.1186/s13636-019-0160-1.

3. Yuki Takashima, Ryoichi Takashima, Tetsuya Takiguchi, and Yasuo Ariki: "Knowledge Transferability between the Speech Data of Persons with Dysarthria Speaking Different Languages for Dysarthric Speech Recognition," *IEEE Access*, vol. 7, pp. 164320-164326, 2019, doi: 10.1109/ACCESS.2019.2951856.

International Conference Papers

1. Yuki Takashima, Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki: "Feature Extraction Using Pre-Trained Convolutive Bottleneck Nets for Dysarthric Speech Recognition," in *Proc. European Signal Processing Conference (EUSIPCO)*, pp. 1426-1430, Sept. 2015.

0. PUBLICATION LIST

2. Ryo Aihara, Yuki Takashima, Tetsuya Takiguchi, and Yasuo Ariki: "Home Appliance Control Using Speech Recognition for a Person with an Articulation Disorder," in *Proc International Symposium on Applied Electromagnetics and Mechanics (ISEM)*, 2 pages, Sept. 2015.

3. Yiting Li, Yuki Takashima, Tetsuya Takiguchi, and Yasuo Ariki: "Lip Reading Using a Dynamic Feature of Lip Images and Convolutional Neural Networks," in *Proc. IEEE/ACIS International Conference on Computer and Information Science (ICIS)*, pp. 971-976, June 2016.

4. Yuki Takashima, Ryo Aihara, Tetsuya Takiguchi, Yasuo Ariki, Nobuyuki Mitani, Kiyohiro Omori, and Kaoru Nakazono: "Audio-Visual Speech Recognition Using Bimodal-Trained Bottleneck Features for a Person with Severe Hearing Loss," in *Proc. Interspeech*, pp. 277-281, Sept. 2016.

5. Yuki Takashima, Tetsuya Takiguchi, Yasuo Ariki, and Kiyohiro Omori: "Audio-Visual Speech Recognition for a Person with Severe Hearing Loss Using Deep Canonical Correlation Analysis," in *Proc. International Workshop on Challenges in Hearing Assistive Technology (CHAT)*, pp. 77-81, Aug. 2017.

6. Yuki Takashima, Hajime Yano, Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki: "Parallel-Data-Free Dictionary Learning for Voice Conversion Using Non-Negative Tucker Decomposition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5294-5298, April 2018.

7. Yuki Takashima, Tetsuya Takiguchi, and Yasuo Ariki: "Exemplar-based Lip-to-Speech Synthesis Using Convolutional Neural Networks," in *Proc. International Workshop on Frontiers of Computer Vision (IWFCV)*, 5 pages, Feb. 2019.

8. Yuki Takashima, Tetsuya Takiguchi, and Yasuo Ariki: "End-To-End Dysarthric Speech Recognition Using Multiple Databases," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6395-6399, May 2019.

Technical Reports

1. Yuki Takashima, Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki: "Phone Labeling Based on Gaussian Mixture Model for Dysarthric Speech Recognition," *IEICE Technical Report*, vol. 115, no. 99, SP2015-13, pp. 71-76, 2015. (in Japanese)

2. Yuki Takashima, Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki: "Feature Extraction Using Adaptive Restricted Boltzmann Machine for Dysarthric Speech Recognition," *IEICE Technical Report*, vol. 116, no. 477, SP2016-135, pp. 321-326, 2017. (in Japanese)

3. Yuki Takashima, Tetsuya Takiguchi, and Yasuo Ariki: "Data Augmentation Using Multiple Databases for End-to-end Dysarthric Speech Recognition," *IEICE Technical Report*, vol. 118, no. 497, SP2018-118, pp. 335-340, 2019. (in Japanese)

4. Yuki Takashima, Ryoichi Takashima, Tetsuya Takiguchi, and Yasuo Ariki: "Knowledge Transferability between Persons with Dysarthria Speaking Different Languages for Dysarthric Speech Recognition," *IEICE Technical Report*, vol. 119, no. 250, SP2019-25, pp. 45-50, 2019. (in Japanese)

Domestic Conference Papers

1. Yuki Takashima, Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki: "Phoneme Estimation using Deep Boltzmann Machine," *Acoustical Society of Japan 2015 Spring Meeting*, 1-1-2, pp. 3-6, 2015. (in Japanese)

2. Yuki Takashima, Yasuhiro Kakihara, Ryo Aihara, Tetsuya Takiguchi, Yasuo Ariki, Nobuyuki Mitani, Kiyohiro Omori, Kaoru Nakazono: "Convolutional Bottleneck Networks を用いた重度難聴者のマルチモーダル音声認識," *Meeting on Image Recognition and Understanding (MIRU)*, OS3-2, 2 pages, 2015. (in Japanese)

3. Yuki Takashima, Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki: "Phone Labeling based on the Probabilistic Representation for Dysarthric Speech Recognition," *Acoustical Society of Japan 2015 Autumn Meeting*, 1-8-3, pp. 1243-1246, 2015. (in Japanese)

4. Yuki Takashima, Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki: "Acoustic modeling using restricted Boltzmann machine considering speaker and noise," *Acoustical Society of Japan 2016 Spring Meeting*, 3-P-10, pp. 175-178, 2016. (in Japanese)

5. Yuki Takashima, Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki: "Multi-modal speech recognition using factored 3-way restricted Boltzmann machine," *Acoustical Society of Japan 2016 Autumn Meeting*, 3-Q-10, pp. 109-112, 2016. (in Japanese)

6. Yuki Takashima, Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki: "Dysarthric speech recognition using Gaussian-Gaussian RBM," *Acoustical Society of Japan 2017 Spring Meeting*, 1-Q-5, pp. 95-98, 2017. (in Japanese)

7. Yuki Takashima, Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki: "Audio-Visual Speech Recognition for a Person with Severe Hearing Loss Using Deep Canonical Correlation Analysis," *Acoustical Society of Japan 2017 Autumn Meeting*, 1-R-24, pp. 119-122, 2017. (in Japanese)

8. Yuki Takashima, Hajime Yano, Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki: "Parallel-Data-Free Dictionary Learning for Voice Conversion Using Non-negative Tucker Decomposition," *Acoustical Society of Japan 2018 Spring Meeting*, 1-9-3, pp. 211-214, 2018. (in Japanese)

9. Yuki Takashima, Tetsuya Takiguchi, and Yasuo Ariki: "ハイスピードカメラ画像を用いた唇動画像からの音声生成," *Meeting on Image Recognition and Understanding (MIRU)*, PS1-65, 2 pages, 2018. (in Japanese)

10. Yuki Takashima, Tetsuya Takiguchi, and Yasuo Ariki: "High-frequency Production Based on Non-negative Matrix Factorization for Articulation

Disorders," *Acoustical Society of Japan 2018 Autumn Meeting*, 2-5-5, pp. 1309-1312, 2018. (in Japanese)

11. Yuki Takashima, Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki: "End-to-End dysarthric speech recognition using multiple databases," *Acoustical Society of Japan 2019 Spring Meeting*, 2-9-4, pp. 869-872, 2019. (in Japanese)

12. Yuki Takashima, Ryo Aihara, Ryoichi Takashima, Tetsuya Takiguchi, and Shu Murayama: "Cross-modal Teacher-Student Learning for Lip reading," *Acoustical Society of Japan 2019 Autumn Meeting*, 2-3-5, pp. 823-826, 2019. (in Japanese)

Glossary

AAM	Active Appearance Model
ARBM	Adaptive Restricted Boltzmann Machine
ASM	Active Shape Model
ASR	Automatic Speech Recognition
BN	BottleNeck
CBN	Convolutive Bottleneck Network
CC	Cepstral Coefficients
CLM	Constrained Local Model
CNN	Convolutional Neural Network
CTC	Connectionist Temporal Classification
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DNN	Deep Neural Network
DP	Dynamic Programing
DPM	Dynamic Programming Matching
DTW	Dynamic Time Warping
E2E	End-to-End
EM	Expectation Maximization
FFT	Fast Fourier Transform
GAN	Generative Adversarial Network
GBRBM	Gaussian-Bernoulli Restricted Boltzmann Machine
GMM	Gaussian Mixture Model
GV	Global Variance
HMM	Hidden Markov Models
IDFT	Inverse Discrete Fourier Transform
JDGMM	joint density Gaussian Mixture Model

LAS	Listen, Attend and Spell
LDA	Linear Discriminant Analysis
LFCC	Linear Frequency Cepstral Coefficient
LSTM	Long Short-Term Memory
LPC	Linear Prediction Coefficients
MAF	Multiple Acoustic Frame
MCD	Mel-Cepstral Distortion
MFCC	Mel-Frequency Cepstral Coefficients
MGC	Mel Generalized Cepstrum
MLP	Multi-Layer Perceptrons
NN	Neural Network
NMF	Non-negative Matrix Factorization
MOS	Mean Opinion Score
NTD	Non-negative Tucker Decomposition
PCA	Principal Component Analysis
PDM	Point Distribution Model
PER	Phoneme Error Rate
pLSTM	Pyramid bidirectional Long Short-Term Memory
RBM	Restricted Boltzmann Machine
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
TTS	Text-To-Speech
VC	Voice Conversion
VCC	Voice Conversion Challenge

Chapter 1

Introduction

1.1 Background

The person with an articulation disorder has problems forming speech sounds properly. Although the voice is one of the most natural communication tools for human beings, the voices of persons with articulation disorders are difficult to understand for people around them. Because they do not have an intellectual disability, they can act with definite intent. If making communication go smoothly is realized, the quality of their life is dramatically improved. Therefore, there is a critical need for voice-driven assistive systems. In this paper, we focus on persons with articulation disorder resulting from the athetoid type of cerebral palsy and the severe hearing loss.

Cerebral palsy results from damage to the central nervous system, and the damage causes movement disorders. In the case of a person with this type of articulation disorder, his/her movements are sometimes more unstable than usual [1]. That means their utterances (especially their consonants) are often unstable or unclear due to the athetoid symptoms. Athetoid symptoms also restrict the movement of their arms and legs. Most people suffering from athetoid cerebral palsy cannot communicate by sign language or writing, so there is great need for systems to understand their speech [2].

Severe hearing loss refers that speech is inaudible without hearing aids or cochlear implants. Although the persons without hearing loss can control the volumes of their voices and their speaking style in a noisy environment, those with

hearing loss cannot do it, because they cannot hear ambient sound. Therefore, it is difficult for people around them to recognize utterances using only the speech signal. Reading lips ("lip reading") is one communication skill that can help some of the persons with severe hearing loss communicate better. People who can use lip reading as the communication tool can make the proper lip shape [3]. So, the lip movement of the speaker can be used to compensate for utterance recognitiont [4].

There are three approaches to assistive technology for a person with an articulation disorder. The first is text-to-speech (TTS) that is a technique for converting the given text into human-like speech. In this approach, the user must type in what you want to say. However, in the case of the person with an articulation disorder, it is difficult to use the keyboard or cellular phone owing to athetoid symptoms. The second is voice conversion (VC) that is a technique for converting the specific information in speech while preserving the other information. If we convert dysarthric speech into non-dysarthric speech, we can use the VC method to obtain a converted voice that has high intelligibility [5]. On the other hand, if we convert non-dysarthric speech into dysarthric speech, we can use the converted speech as an additional speech data to train a model for various tasks [6]. The third is automatic speech recognition (ASR) that is a technique for converting the speech signal into its corresponding sequence of words or other linguistic entities. In this approach, we can visually understand disordered speech as the corresponding text. The recognized transcription can also be used for an input text on TTS. Because we consider that there is great benefit for the voice-driven systems, this paper adopts the VC and ASR approaches.

Most existing works to develop VC and ASR systems require a large amount of training data. However, it is difficult to collect such data of persons with articulation disorders owing to a physical burden. Therefore, a method that can train models from the limited amount of training data is needed. Moreover, their speech style differs significantly from that of physically-unimpaired persons. A speaker-independent ASR system trained using physically unimpaired persons hardly recognize their speech. The model should have a mechanism to learn dysarthric characteristics effectively. This paper focuses on an approach specifically tailored to overcome these problems of impaired speech.

1.2 Approaches

For the assistive technology of dysarthric speech, this paper introduces two approaches: conversion and recognition.

1.2.1 Conversion-based approach

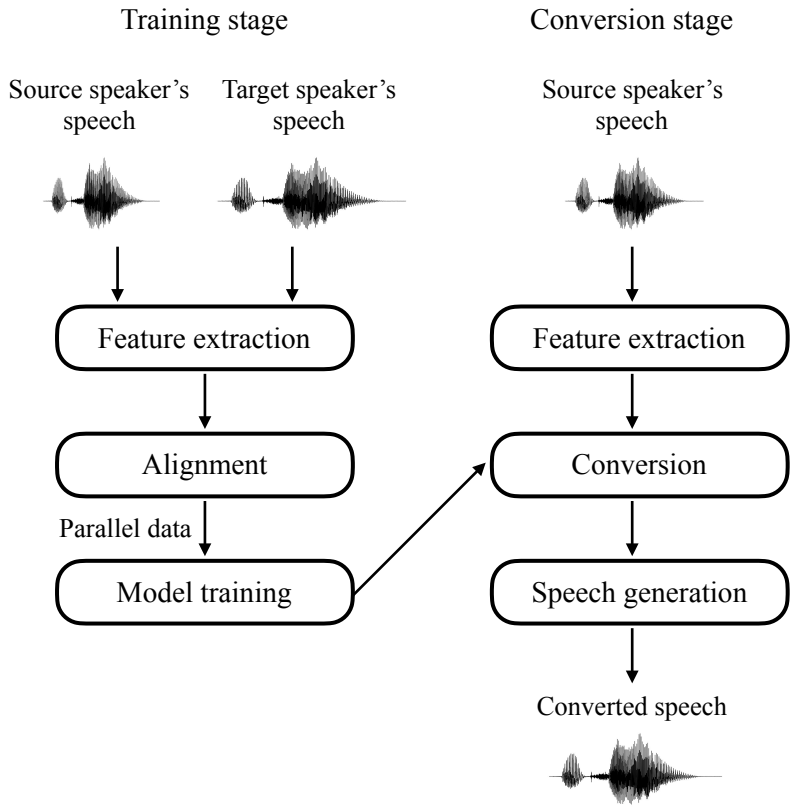


Figure 1.1: Flowchart of a typical VC system

Figure 1.1 shows a system flow of most VC systems. The system can broadly be divided into two stages: training and conversion. Both stages begin with feature extraction. In this process, acoustic features, such as mel-frequency cepstral coefficients (MFCC), cepstral coefficients (CC), linear prediction coefficients (LPC) , or mel generalized cepstrum (MGC) are extracted from the speech signals. In the training stage, the extracted time-series features are aligned to adjust the time positions of the source speaker's features and target speakers' features

as necessary. Such aligned data is called "parallel data" that is a pair of speech signals to be produced by the source and the target speakers uttering the same sentence. Dynamic time warping (DTW) [7] is often used for the alignment process. In the conversion stage, the acoustic features are extracted from the source speaker's speech and fed to the model, resulting in acoustic features that are supposed to be those of the target speakers. Finally, the features are back-projected into a speech signal. Then we obtain converted speech.

Various statistical approaches of VC have been studied so far as discussed in [8, 9]. Among these approaches, the Gaussian mixture model (GMM)-based mapping method [10] is most widely used, and a number of improvements have been proposed [11, 12, 13]. Other VC methods, such as approaches based on non-negative matrix factorization (NMF) [14, 15], neural networks (NNs) [16], and restricted Boltzmann machines (RBMs) [17, 18], have been also proposed. In this paper, we employ NMF-based VC. This approach can provide a natural-sounding converted voice [19] by avoiding over-smoothing and over-fitting problems reported in [12] that used the other statistical approaches such as GMM-based VC.

NMF [20] is one of the most popular sparse representation methods. NMF decomposes the input observation into two matrices — the basis matrix and weight matrix. The goal of NMF is to estimate these two matrices from the input observation. In this paper, we refer to the basis matrix as the "dictionary" and the weight matrix as the "activity". It is assumed that the dictionary and the activity represent the speaker identity and the phonemic information, respectively. By replacing the source speaker dictionary with that of the target speaker, the original speech spectrum is converted to that of the target speaker. If we assume that the dictionary also represents the dysarthric characteristic, VC can be used for the conversion between dysarthric and non-dysarthric speech. The NMF-based VC method results in a more natural-sounding converted voice [19] compared with conventional VC methods.

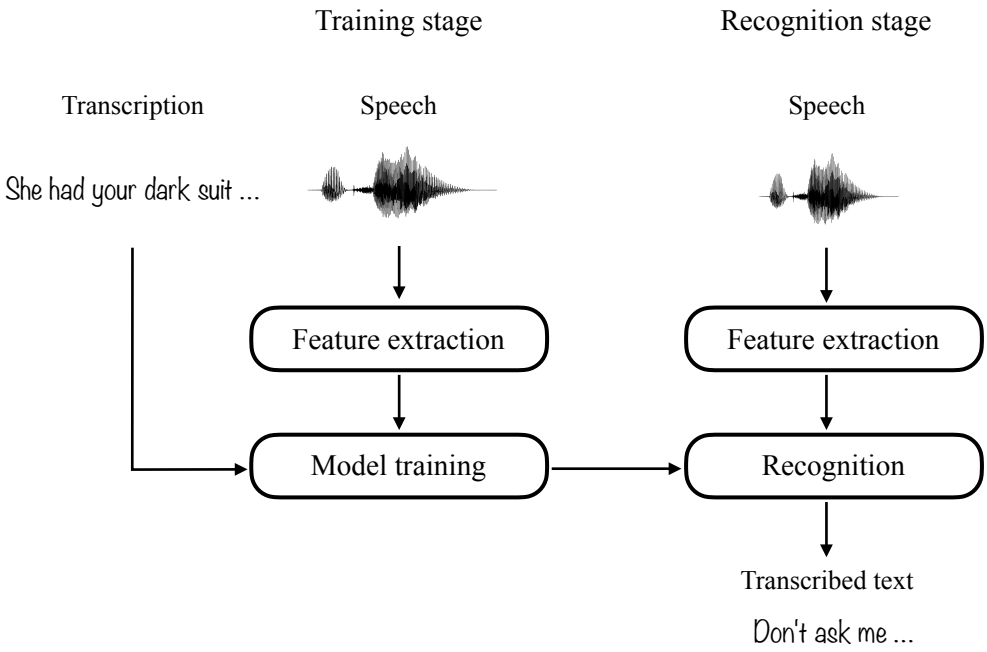


Figure 1.2: Flowchart of a typical ASR system

1.2.2 Recognition-based approach

Figure 1.2 shows a system flow of a typical ASR system. The system can broadly be divided into two stages: training and recogntion. Both stages begin with feature extraction, as well as VC. In the training state, the speech signal and the associated transcription are required. In the recognition stage, unknown utterances are transcribed using the trained model.

The main part in ASR is acoustic pattern matching that detect and classify possible acoustic pattern from the acoustic feature. The most important progress has been achieved using techniques based on the DTW, hidden Markov models (HMM), and neural networks (NN). In the DTW-based ASR system [21], given the input acoustic pattern, the system seeks the best matching pattern from the templates while minimizing errors between the input pattern and each template. Although DTW is a non-parametric technique, it has been shown that DTW can be considered as a special case of HMM modeling, which is a parametric technique and improves recognition accuracy [22, 23]. In line with the progress of deep learning, deep neural networks (DNN) was introduced to estimate the pos-

terior probability of an HMM state [24, 25, 26]. More recently, end-to-end (E2E) ASR has gained popularity [27, 28]. This system folds all the modules of the ASR system into a single NNs. In this paper, we employ E2E ASR.

In recent years, E2E ASR shows promising results in various tasks with its extremely simplified training and decoding schemes [29, 30]. Unlike traditional HMM-based ASR systems, E2E ASR models learn all the components of the ASR system jointly. Therefore, it is easy to develop ASR systems for new applications and configurations. Moreover, the NNs can represent the complex nonlinear transformation by stacking the hidden layers that have different functions. We assume that the E2E ASR approach is effective against atypical speech such as disordered speech.

1.3 Purpose of This Thesis

To achieve an ASR system that works well for dysarthric speech, in this paper, we tackle the following two problems: the limited data availability and the difference in a speech style. This paper proposes one VC algorithm and two different ASR algorithms for each practical task. All our methods focus on the speech data of persons with two types of articulation disorders: the cerebral palsy and the severe hearing loss. The proposed methods correspond to the following key words: parallel-data-free, transfer learning, and multimodality. Fig. 1.3 shows the relationships of our proposed method.

1.3.1 Three Proposed systems

1.3.1.1 Parallel-data-free VC

A person with an articulation disorder may not be able to utter the given text clearly due to athetoid symptoms. In conventional NMF-VC, models are trained using parallel data which results in the speech data requiring elaborate preprocessing to generate parallel data. NMF-VC also tends to be an extensive model as this method has several parallel exemplars for the dictionary matrix, leading to a high computational cost. In Chapter 4, an innovative parallel dictionary-learning method using non-negative Tucker decomposition (NTD) is proposed.

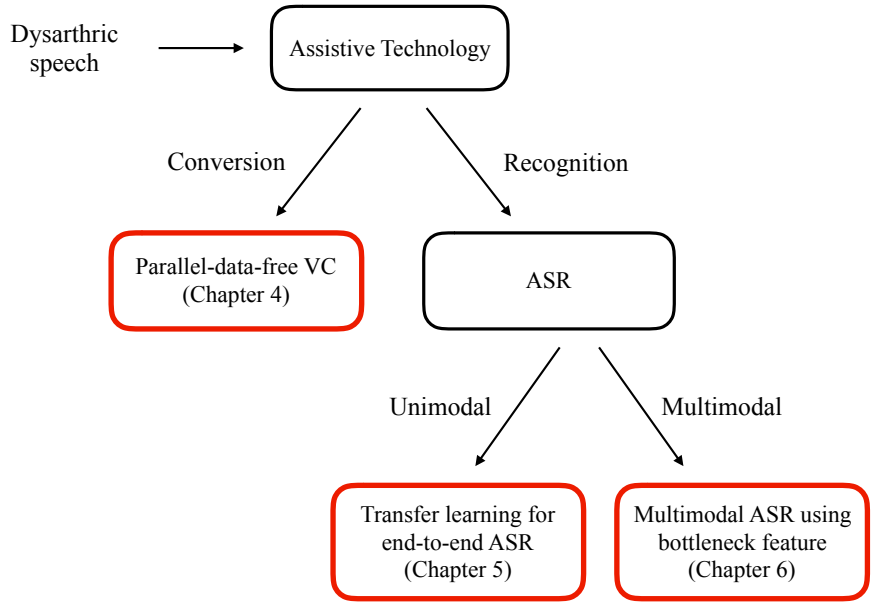


Figure 1.3: Flow of this thesis

The proposed method uses tensor decomposition and decomposes an input observation into a set of mode matrices and one core tensor. The NTD-based dictionary-learning method estimates the dictionary matrix for NMF-VC without using parallel data. The effectiveness of our method is confirmed in both parallel and nonparallel settings.

1.3.1.2 Transfer learning for ASR

In Chapter 5, we present an E2E speech recognition system for Japanese persons with articulation disorders resulting from athetoid cerebral palsy. Because their utterance is often unstable or unclear, speech recognition systems struggle to recognize their speech. Recent deep learning-based approaches have exhibited promising performance. However, these approaches require a large amount of training data, and it is difficult to collect sufficient data from such dysarthric people. We proposes a transfer learning method that transfers two types of knowledge corresponding to the different datasets: the language-dependent (phonetic and linguistic) characteristic of unimpaired speech and the language-independent characteristic of dysarthric speech. The former is obtained from Japanese non-

dysarthric speech data, and the latter is obtained from non-Japanese dysarthric speech data. In the proposed method, we pre-train a model using Japanese non-dysarthric speech and non-Japanese dysarthric speech, and thereafter, we fine-tune the model using the target Japanese dysarthric speech. To handle the speech data of the two different languages in one model, we employ language-specific decoder modules. Experimental results indicate that our proposed approach can significantly improve speech recognition performance compared with other approaches that do not use additional speech data.

1.3.1.3 Multimodal ASR using bottleneck feature

An audio-visual (multimodal) speech recognition system for a person with an articulation disorder resulting from severe hearing loss is proposed in Chapter 6. Similar to articulation disorder resulting from athetoid cerebral palsy, the speech style of a person with this type of articulation disorder is also quite different from those of people without hearing loss, thus a speaker-independent model for unimpaired persons is hardly useful for recognizing his/her voice. In addition, the persons with hearing loss cannot control the volumes of their voices and their speaking style even when there is a lot of background noise, because they cannot recognize that noise. Therefore, to recognize their speech is especially difficult in a noisy environment. The lip shape is an important interface for communication for some persons with hearing loss. We investigate in this study an audio-visual speech recognition system in noisy environments, where a robust feature extraction method using a convolutive bottleneck network (CBN) is applied to audio-visual data. The bottleneck feature extracted from CBN is a compact representation that aggregates the phone-associated information. We confirmed the effectiveness of this approach through word-recognition experiments in noisy environments, where the CBN-based feature extraction method outperformed the conventional methods.

1.3.2 Novelties of This Thesis

This paper proposes new algorithms for three practical tasks.

In NMF-based VC, parallel-data-free conversion is achieved. Many expansion of NMF-based VC based on parallel data has been proposed [31, 32]. Our proposed method will also release the constraint of the data structure for these methods.

The use of multiple databases is proposed for dysarthric ASR. In fields with access to a limited amount of training data, our approach enables to obtain a well-trained model. This approach is also useful as a data augmentation in such fields.

By using the visual modality, the recognition accuracy is compensated effectively. Experimental results show the multimodal approach is effective in the noisy environment. Our method can be expanded using not only the lip images but also the cheek or throat images.

1.4 Outline

Starting in Chapter 2, we list some feature extraction methods related to speech processing. In Chapter 3, basics of VC and ASR are described. In Chapter 4, parallel-data-free VC is described. Chapter 5 describes a novel framework of transfer learning that uses multiple databases. In Chapter 6, multimodal ASR using the bottleneck feature is described. Finally, Chapter 7 concludes the thesis.

Chapter 2

Acoustic Features

In this chapter, we review common signal processing used in voice conversion (VC) and automatic speech recognition (ASR). First, we explain short-time Fourier analysis that is a kind of linear time-frequency analysis to obtain a localized spectrum in time domain. Second, cepstrum analysis, which is a fundamental method to extract important information from a speech signal, is presented. Third, we discuss MFCC, which is a well-known feature in speech signal processing. Finally, an analyzing speech synthesis method using vocoder is presented.

2.1 Speech signal processing

2.1.1 Short-time Fourier analysis

Short-time Fourier analysis is a method for analysing time-varying waveforms in the frequency domain. A number of fundamental concepts and definitions of this analysis can be found in [33, 34]. A typical speech processing system extracts an acoustic feature based on short-time Fourier analysis.

Short-time analysis depends on windowing of the speech signal to isolate a short-time interval for spectral analysis. The short-time analysis interval is called a *frame*, and the length of the frame is called the *frame length*. The windowing proceeds along the time axis by shifting an appropriate interval to represent the temporal dynamic feature. The shifting interval is called the *frame shift*.

Let a continuous speech signal be denoted $s(t)$ and the window function by $w(t-\tau)$, the signal after windowing is given as:

$$x(t,\tau) = s(t)w(t-\tau), \tag{2.1}$$

where τ is a time when the window is applied. $x(t,\tau)$ is a signal as a function of time t with the window position τ.

In practice, the continuous time signal $x(t,\tau)$ are quantized by sampling and digitizing for computer processing. Suppose that the continuous signal $x(t,\tau)$ is sampled at a sampling period of T seconds, the discrete sampled data results in $x_m(n)$, where n indicates discrete time and m corresponds to the time when a window is applied. The corresponding discrete Fourier transform (DFT) of a discrete sample sequence $\{x_m(n)\}$ is defined as:

$$X_{m,k} = \sum_{n=0}^{N-1} x_m(n)\mathrm{e}^{-j\frac{2\pi}{N}kn} \tag{2.2}$$

where N and k are the number of sampled data to be analyzed (frame length) and an index of the discrete frequency, respectively. This operation is called a short-time Fourier transform. The DFT transforms a sequence of the real numbers into a sequence of the complex numbers, which is defined as:

$$X_{m,k} = |X_{m,k}| \exp(j\arg[X_{m,k}]), \tag{2.3}$$

where $|X_{m,k}|$ is the magnitude/amplitude spectrum, and $\arg[X_{m,k}]$ is the phase spectrum. $|X_{m,k}|^2$ is the power spectrum. The inverse discrete Fourier transform (IDFT) is defined as:

$$\tilde{x}_m(n) = \frac{1}{N}\sum_{k=0}^{N-1} X_{m,k}\mathrm{e}^{j\frac{2\pi}{N}kn} \tag{2.4}$$

A fast algorithm for computation of the DFT is called an fast Fourier transform (FFT) and is normally applicable where N is a power of 2.

2.1.2 Cepstrum analysis

The speech signal is generated from vibrations of the vocal cord. The vibration passes the vocal tract of the speaker and arrives at the listener's ears or a microphone. When we vibrate our vocal chords and change the shape of our mouth and vocal tracts, the sounds of various phonemes such as /a/ or /i/ can be generated. This speech generating process is modeled as a source-filter model (Figure 2.1). The filter associated with the vocal tract is called the formant and the fundamental frequency from the vibration is called the pitch. In speech signal processing, the information related to the formant that determines phonemes is reasonably important. Therefore, when the system recognizes a given speech, we obtain better results with the formant information than with the original spectrum.

One well-known technique for extracting formants is cepstrum analysis [35]. The cepstrum is obtained as follows: 1) execute Fourier transform to the given speech, 2) take absolute and logarithm, and 3) execute inverse Fourier transform.

The observed speech signal results in a convolution of the vocal cord (the excitation) and the vocal tract impulse response in the time domain. By applying the spectrum analysis, its spectrum is the product of the vocal cord and the filter spectra in the frequency domain. Letting $G(\omega)$ and $H(\omega)$ be the spectrum of the vocal cord and the tract, respectively, the spectrum of the speech $S(\omega)$ is represented as

$$S(\omega) = G(\omega) \cdot H(\omega). \tag{2.5}$$

When we apply a logarithm and inverse Fourier transform to Eq. (2.5), we obtain

$$\log|S(\omega)| = \log|G(\omega)| + \log|H(\omega)| \tag{2.6}$$

and

$$\hat{c}(d) = DFT^{-1}\{\log|S(\omega)|\} \tag{2.7}$$

$$= DFT^{-1}\{\log|G(\omega)|\} + DFT^{-1}\{\log|H(\omega)|\}, \tag{2.8}$$

where $\hat{c}(d)$ indicates the d-th cepstrum (d is quefrency axis).

When we regard each spectrum in Figure 2.1 as a signal, we notice that the vocal cord $G(\omega)$ signal changes rapidly, and the vocal tract $H(\omega)$ signal

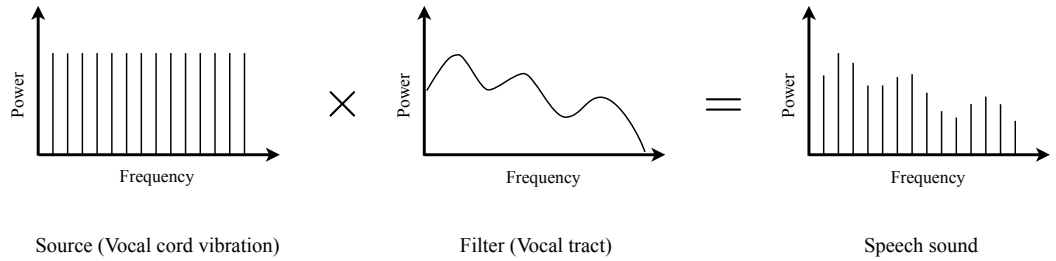


Figure 2.1: Source filter model: speech sound can be generated by multiplying spectra of the vocal cord vibration (source) and spectra of the vocal tract (filter).

changes slowly by contrast. By applying an inverse Fourier transform to these signals, $G(\omega)$ locates at high quefrency owing to its periodic high frequency, and $H(\omega)$ locates at low quefrency owing to its smoothed spectral envelope. This separable representation is very suitable to the deconvolution of speech and this analysis is called cepstral analysis. Therefore, the vocal tract information $DFT^{-1}\{\log|H(\omega)|\}$ can be extracted by liftering (low-pass filtering in quefrency) as $\hat{c}_l(d)$. To obtain the spectral envelope, we invert $\hat{c}_l(d)$ into the log amplitude spectrum by DFT and then calculate the exponential function of it. Figs. 2.2 and 2.3 show the flow and results of the cepstral analysis.

2.1.3 MFCC

The cepstrum feature introduced in the previous section was obtained as a linear log spectrum and is called the linear frequency cepstral coefficient; LFCC). On the other hand, there is another formant extraction method, MFCC [36] is extracted from the transformation on the mel-scale, which approximates the human auditory scale sensitive to pitch.

When we, human beings, hear something, auditory sensitivity becomes poorer as pitch increases. That means that the frequency resolution is very low at high frequencies and high at low frequencies. The relationship is not linear but nonlinear, as approximated by

$$f' = 1127.01048 \log(1 + \frac{f}{700}), \tag{2.9}$$

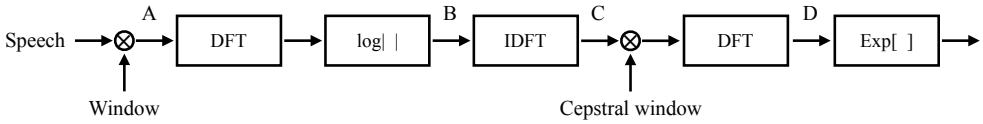
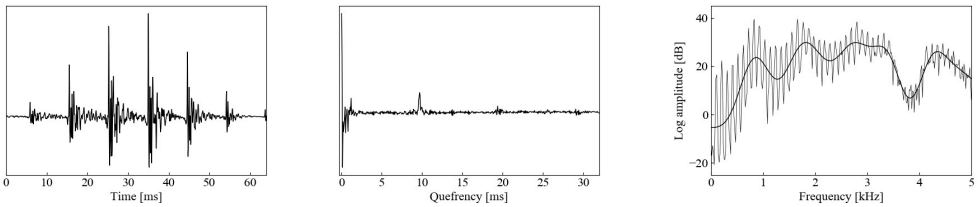


Figure 2.2: The flow of the cepstral analysis.



(a) Windowed speech signal (b) Cepstrum of the voiced sound (c) Log amplitude spectrum

Figure 2.3: Result of the cepstral analysis for the voiced sound. (a) the windowed speech signal following pre-emphasis (corresponds to A in Fig 2.2), (b) the cepstrum (corresponds to C in Fig 2.2), (c) the log amplitude spectrum and the spectral envelope (correspond to B and D in Fig 2.2).

where f' is the mel frequency. Typically, we use filter-bank representation instead of using the coefficients themselves. In the MFCC approach, mel-filter-banks as shown in Figure 2.4 are used. Each filter is triangular, and outputs the sum value of the multiplication. The power spectrum of the mel-scale frequency $M(i)$ is obtained by

$$M(i) = \sum_{f=f'_{i-1}}^{f'_{i+1}} W_i(f) \cdot |X(f)|^2. \tag{2.10}$$

From Eq. (2.10), the MFCC can be calculated as follows:

$$M_{cep}(d) = DFT^{-1}\{\log M(i)\}. \tag{2.11}$$

MFCC contains some useful information on representing speech using a small numbers of dimensions, therefore most works in speech signal processing, specifically in speech recognition, deal with these features. However, as the MFCC is based on a filter-bank calculation, it is, in general, difficult to reconstruct the speech signal from the obtained MFCC because the values in a filter-bank are summed up with the weights. Milner and Shao proposed an approximation approach for speech reconstruction from MFCC using a source-filter model [37].

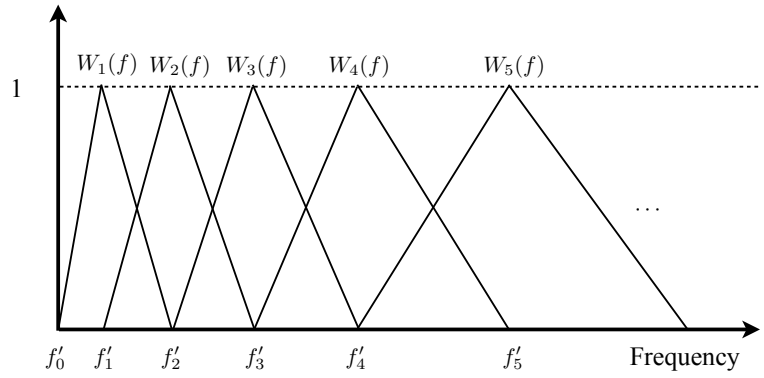


Figure 2.4: Mel-scale filter banks

Other approaches such as using a z-transform [38, 39] and using a mel-log spectral approximation (MLSA) filter [40] have also been proposed.

2.1.4 Vocoder

Vocoder is an analysis and synthesis method for speech signal processing. This paper uses WORLD [41], which is provided by Morise and is written as C++ and MATLAB codes. This tool decomposes a speech signal into three parameters: fundamental frequency (F_0), spectral envelope and aperiodicity. The analysis and synthesis quality is fairly high, so many researchers in speech signal processing use it. Figure 2.5 depicts the common way of using WORLD to modify a speech signal. The spectral envelope extracted using WORLD resembles formant information in cepstrum analysis. Therefore, in voice conversion, we usually modify the spectrum envelope and the F_0 to the desired ones; we do not change the aperiodic parameters at the synthesis stage. We can also extract MFCC features from the WORLD spectrum.

Because the vocoder is designed for the physically unimpaired person, speech uttered by persons with articulation disorders may cause the misestimation of parameters. Although the vocoder is a convenient tool for speech analysis and synthesis, estimated parameters should be carefully used.

STRIAGHT [42, 43] is a vocoder provided by Kawahara. In [44], Kawahara proposed advanced version of STRIAGHT, which is named as "TANDEM-

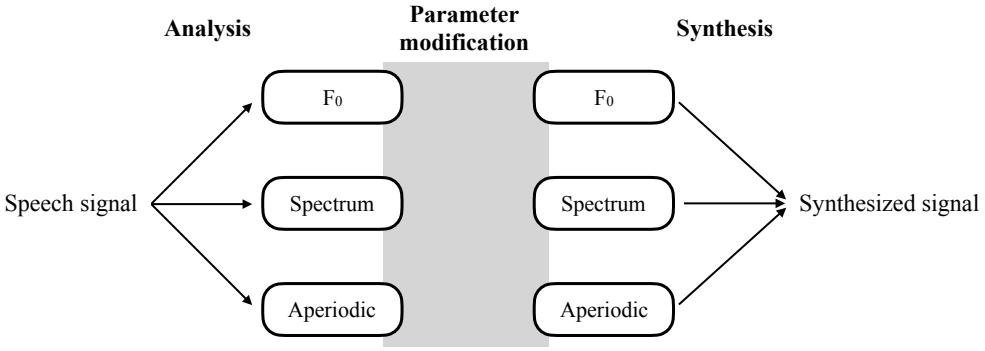


Figure 2.5: Modifying a speech signal using WORLD.

STRAIGHT". WORLD was proposed to reduce the computational cost of TANDEM-STRAIGHT without deterioration. Ahocoder [45] is also used in some VC and text-to-speech application.

Chapter 3

Conventional methods

In this chapter, we review the conventional voice conversion (VC) and automatic speech recognition (ASR) methods. First, conventional statistical VC methods are explained [46, 47]. Next, conventional ASR methods are described [29, 48].

3.1 Conventional VC Methods

In order to estimate the models, most existing VC approaches need sets of speech signals where the same sentences are uttered by a source and a target speaker. Furthermore, the training data must be aligned at the frame level. The aligned data is called parallel data, and is often obtained using dynamic time warping (DTW) or dynamic programming (DP) [7, 49, 50]. Such feature vectors are used for training each model. Let us refer to the D-dimensional feature vectors in each frame of the source and target speakers as $\mathbf{x}$ and $\mathbf{y}$, respectively. Assuming the parallel data includes T frames, the training data consists of the source speaker's set $\mathbf{X} \ni \{\mathbf{x}_t\}_{t=1}^{T}$ and the target speaker's set $\mathbf{Y} \ni \{\mathbf{y}_t\}_{t=1}^{T}$.

3.1.1 GMM-based VC

3.1.1.1 Gaussian Mixture Model (GMM)

A Gaussian mixture model (GMM) is a statistical probabilistic model for representing observed data that can be categorized into sub-components. Each component is represented as a multivariate Gaussian distribution $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with

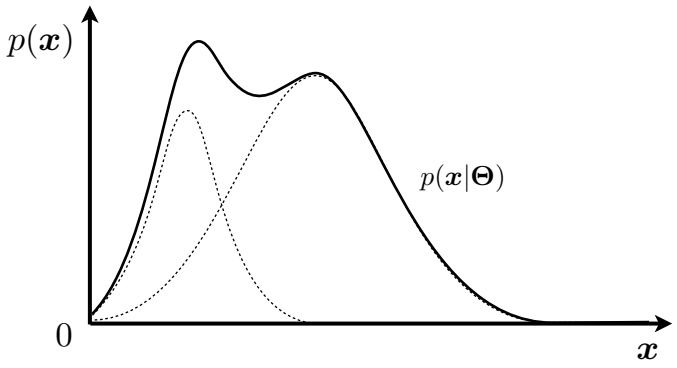


Figure 3.1: Example of a Gaussian mixture model ($M = 2$).

parameters of a D-dimensional mean vector $\boldsymbol{\mu}$ and a variance matrix $\boldsymbol{\Sigma}$, defined as

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\mathsf{T} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right), \tag{3.1}$$

where $\mathbf{x}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ have the elements as follows:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix}, \quad \boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_D \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \cdots & \sigma_{1D}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & & \vdots \\ \vdots & & \ddots & \vdots \\ \sigma_{D1}^2 & \cdots & \cdots & \sigma_{DD}^2 \end{bmatrix}, \tag{3.2}$$

and $(\cdot)^\mathsf{T}$ and $|\cdot|$ indicate the transpose and determinant of the matrix, respectively.

GMM represents the overall distribution of the data using a weighted sum of the components. The probability density function (pdf) of GMM is defined as

$$p(x|\boldsymbol{\Theta}) = \sum_{m=1}^{M} \alpha_m \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m), \tag{3.3}$$

where M indicates the number of mixtures. $\boldsymbol{\Theta}$ is a set of parameters of GMM, which contains α_m, $\boldsymbol{\mu}_m$, and $\boldsymbol{\Sigma}_m$ for all m.

Figure 3.1 shows an example of a one-dimensional GMM that has two components, depicted by a solid line. The GMM was obtained from the weighted sum of the two Gaussian distributions, depicted by dotted lines.

GMM parameters can be estimated using the expectation maximization (EM) algorithm [51]. The algorithm repeats E-step (expectation) and M-step (maximization) by turns. First, all parameters are randomly initialized. In the E-step,

we calculate a Q-function (expectation of log-likelihood) defined as

$$\begin{aligned} Q(\hat{\boldsymbol{\Theta}}|\boldsymbol{\Theta}) &= E[\log p(\mathbf{x}, m|\hat{\boldsymbol{\Theta}})]_{p(m|\mathbf{x},\boldsymbol{\Theta})} \\ &= \sum_{m=1}^{M} p(m|\mathbf{x}, \boldsymbol{\Theta}) \log p(\mathbf{x}, m|\hat{\boldsymbol{\Theta}}), \end{aligned} \tag{3.4}$$

where

$$\begin{aligned} p(\mathbf{x}, m|\hat{\boldsymbol{\Theta}}) &= \prod_{n=1}^{N} p(\mathbf{x}_n, m|\hat{\boldsymbol{\Theta}}) \\ &= \prod_{n=1}^{N} \hat{\alpha}_m \mathcal{N}(\mathbf{x}_n; \hat{\boldsymbol{\mu}}_m, \hat{\boldsymbol{\Sigma}}_m). \end{aligned} \tag{3.5}$$

Therefore, Eq. (3.4) becomes

$$\begin{aligned} Q(\hat{\boldsymbol{\Theta}}|\boldsymbol{\Theta}) = &\sum_{k}\sum_{n} p(m = k|\mathbf{x}_n, \boldsymbol{\Theta}) \log \hat{\alpha}_k \\ &+ \sum_{k}\sum_{n} p(m = k|\mathbf{x}_n, \boldsymbol{\Theta}) \log \mathcal{N}(\mathbf{x}_n; \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k). \end{aligned} \tag{3.6}$$

In the M-step, update rules for each parameter are derived to maximize the Q-function (Eq. (3.6)). The derived update rules are

$$\hat{\alpha}_k = \frac{1}{N} \sum_{n=1}^{N} p(k|\mathbf{x}_n, \boldsymbol{\Theta}), \tag{3.7}$$

$$\hat{\boldsymbol{\mu}}_k = \frac{\sum_n p(k|\mathbf{x}_n, \boldsymbol{\Theta}) \mathbf{x}_n}{\sum_n p(k|\mathbf{x}_n, \boldsymbol{\Theta})}, \tag{3.8}$$

$$\hat{\boldsymbol{\Sigma}}_k = \frac{\sum_n p(k|\mathbf{x}_n, \boldsymbol{\Theta})(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_k)^T}{\sum_n p(k|\mathbf{x}_n, \boldsymbol{\Theta})}, \tag{3.9}$$

where $p(k|\mathbf{x}_n, \boldsymbol{\Theta})$ is the probability that $\mathbf{x}_n$ is sampled from the k-th component, which is calculated by

$$p(k|\mathbf{x}_n, \boldsymbol{\Theta}) = \frac{\alpha_k \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_k \alpha_k \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}. \tag{3.10}$$

3.1.1.2 Conversion Based on Maximum Likelihood Estimation

When it comes to voice conversion based on GMM, we modeled a joint probability of the source and the target speakers using GMM. Therefore, this model is called

the joint density GMM (JD-GMM). In the training stage of the JD-GMM, we use a joint vector $\mathbf{Z}$ that concatenates the source speakers vector $\mathbf{X} = [\mathbf{x}^\mathsf{T} \Delta \mathbf{x}^\mathsf{T}]^\mathsf{T}$ and target speakers vector $\mathbf{Y} = [\mathbf{y}^\mathsf{T} \Delta \mathbf{y}^\mathsf{T}]^\mathsf{T}$. (i.e., $\mathbf{Z} = [\mathbf{X}^\mathsf{T} \mathbf{Y}^\mathsf{T}]^\mathsf{T}$). The probability $p(\mathbf{Z})$ is modeled using GMM as follows:

$$p(\mathbf{Z}|\boldsymbol{\Theta}^{(z)}) = \sum_{m=1}^{M} \alpha_m \mathcal{N}(\mathbf{Z}; \boldsymbol{\mu}_m^{(z)}, \boldsymbol{\Sigma}_m^{(z)}), \tag{3.11}$$

where $\boldsymbol{\mu}_m^{(z)}$ and $\boldsymbol{\Sigma}_m^{(z)}$ consist of

$$\boldsymbol{\mu}_m^{(z)} = \begin{bmatrix} \boldsymbol{\mu}_m^{(x)} \\ \boldsymbol{\mu}_m^{(y)} \end{bmatrix}, \quad \boldsymbol{\Sigma}_m^{(z)} = \begin{bmatrix} \boldsymbol{\Sigma}_m^{(xx)} & \boldsymbol{\Sigma}_m^{(xy)} \\ \boldsymbol{\Sigma}_m^{(yx)} & \boldsymbol{\Sigma}_m^{(yy)} \end{bmatrix}. \tag{3.12}$$

The parameters $\boldsymbol{\mu}_m^{(x)}$ and $\boldsymbol{\Sigma}_m^{(xx)}$, and the parameters $\boldsymbol{\mu}_m^{(y)}$ and $\boldsymbol{\Sigma}_m^{(yy)}$ correspond to the source speaker's and target speaker's Gaussian distributions, respectively. The parameter $\boldsymbol{\Sigma}_m^{(xy)} (= {\boldsymbol{\Sigma}_m^{(yx)}}^\mathsf{T})$ indicates the covariance matrix between the observed data $\mathbf{X}$ and $\mathbf{Y}$. In voice conversion, we usually use a diagonal matrix for $\boldsymbol{\Sigma}_m^{(xx)}$, $\boldsymbol{\Sigma}_m^{(xy)}$, and $\boldsymbol{\Sigma}_m^{(xx)}$ because full-covariance matrices involve the estimation of a lot of parameters.

At the conversion stage (assuming that the parameters $\boldsymbol{\Theta}^{(z)}$ have already been estimated), we consider the probability of $\mathbf{Y}$ given an input $\mathbf{X}$. That is

$$\begin{aligned} p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\Theta}^{(z)}) &= \sum_{all \mathbf{m}} p(\mathbf{m}|\mathbf{X}, \boldsymbol{\Theta}^{(z)}) p(\mathbf{Y}|\mathbf{X}, \mathbf{m}, \boldsymbol{\Theta}^{(z)}) \\ &= \prod_{t=1}^{T} \sum_{m=1}^{M} p(m|\mathbf{X}_t, \boldsymbol{\Theta}^{(z)}) p(\mathbf{Y}_t|\mathbf{X}_t, m, \boldsymbol{\Theta}^{(z)}) \end{aligned} \tag{3.13}$$

where $\mathbf{m} = \{m_1, m_2, \cdots, m_T\}$ is a mixture component sequence. The probabilities on the right side in Eq. (3.13) are represented as

$$p(m|\mathbf{X}_t, \boldsymbol{\Theta}^{(z)}) = \frac{\alpha_m \mathcal{N}(\mathbf{X}_t; \boldsymbol{\mu}_m^{(x)}, \boldsymbol{\Sigma}_m^{(xx)})}{\sum_{n=1}^{M} \alpha_n \mathcal{N}(\mathbf{X}_t; \boldsymbol{\mu}_n^{(x)}, \boldsymbol{\Sigma}_n^{(xx)})} \tag{3.14}$$

$$p(\mathbf{Y}_t|\mathbf{X}_t, m, \boldsymbol{\Theta}^{(z)}) = \mathcal{N}(\mathbf{Y}_t; \mathbf{E}_{m,t}^{(y|x)}, \mathbf{D}_m^{(y|x)}) \tag{3.15}$$

$$\mathbf{E}_{m,t}^{(y|x)} = \boldsymbol{\mu}_m^{(y)} + \boldsymbol{\Sigma}_m^{(yx)} (\boldsymbol{\Sigma}_m^{(xx)})^{-1} (\mathbf{X}_t - \boldsymbol{\mu}_m^{(x)}) \tag{3.16}$$

$$\mathbf{D}_m^{(y|x)} = \boldsymbol{\Sigma}_m^{(yy)} - \boldsymbol{\Sigma}_m^{(yx)} (\boldsymbol{\Sigma}_m^{(xx)})^{-1} \boldsymbol{\Sigma}_m^{(xy)}. \tag{3.17}$$

A time sequence of the converted feature vector $\hat{\mathbf{y}}$ is determined as follows:

$$\hat{\mathbf{y}} = \arg\max P(\mathbf{Y}|\mathbf{X}, \boldsymbol{\Theta}^{(z)}). \tag{3.18}$$

Eq. (3.18) is performed under the linear conversion between the static feature vectors $\mathbf{y}$ and the static and dynamic feature vectors $\mathbf{Y}$

$$\mathbf{Y} = \mathbf{W}\mathbf{y} \tag{3.19}$$

where $\mathbf{W}$ is a transformation matrix [52].

Eq. (3.13) is approximated with a single mixture component sequence as follows:

$$p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\Theta}^{(z)}) \simeq p(\hat{\mathbf{m}}|\mathbf{X}, \boldsymbol{\Theta}^{(z)}) p(\mathbf{Y}|\mathbf{X}, \hat{\mathbf{m}}, \boldsymbol{\Theta}^{(z)}). \tag{3.20}$$

$\hat{\mathbf{m}}$ denotes the suboptimum mixture component sequence, which is determined as follows

$$\hat{\mathbf{m}} = \arg\max P(\mathbf{m}|\mathbf{X}, \boldsymbol{\Theta}^{(z)}). \tag{3.21}$$

The logarithm of the likelihood function is written as

$$\log p(\mathbf{Y}|\mathbf{X}, \hat{\mathbf{m}}, \boldsymbol{\Theta}^{(z)}) = -\frac{1}{2}\mathbf{Y}^{\mathsf{T}}\mathbf{D}_{\hat{\mathbf{m}}}^{(y|x)^{-1}}\mathbf{Y} + \mathbf{Y}^{\mathsf{T}}\mathbf{D}_{\hat{\mathbf{m}}}^{(y|x)^{-1}}\mathbf{E}_{\hat{\mathbf{m}}}^{(y|x)} + cons \tag{3.22}$$

where

$$\mathbf{E}_{\hat{\mathbf{m}}}^{(y|x)} = [\mathbf{E}_{\hat{m}_1,1}^{(y|x)}, \mathbf{E}_{\hat{m}_2,2}^{(y|x)}, \cdots, \mathbf{E}_{\hat{m}_T,T}^{(y|x)}] \tag{3.23}$$

$$\mathbf{D}_{\hat{\mathbf{m}}}^{(y|x)} = \mathrm{diag}[\mathbf{D}_{\hat{m}_1,1}^{(y|x)}, \mathbf{D}_{\hat{m}_2,2}^{(y|x)}, \cdots, \mathbf{D}_{\hat{m}_T,T}^{(y|x)}]. \tag{3.24}$$

we can estimate the most probable $\hat{\mathbf{y}}$ as follows:

$$\hat{\mathbf{y}} = (\mathbf{W}^{\mathsf{T}}\mathbf{D}_{\hat{\mathbf{m}}}^{(y|x)^{-1}}\mathbf{W})^{-1}\mathbf{W}^{\mathsf{T}}\mathbf{D}_{\hat{\mathbf{m}}}^{(y|x)^{-1}}\mathbf{E}_{\hat{\mathbf{m}}}^{(y|x)}. \tag{3.25}$$

We can also maximize the logarithm of the likelihood function of Eq. (3.13) by employing the EM algorithm ,however, there is little difference in the conversion accuracy between it and the suboptimum mixture component sequence [53].

3.1.1.3 Problems

It is often reported that the GMM-based VC includes over-smoothing problems and over-fitting. The over-smoothing arises because the parameters of multiple Gaussian components are estimated by averaging observations with similar context descriptions. As a result, the outputs distribute near the modes of each component. Over-fitting problems come from this complexity. If we supply more Gaussian components, the model is over-fitted to the training data.

Some methods to tackle the over-smoothing problems have been proposed. Toda *et. al.* proposed a Global Variance (GV) of converted spectra over a time sequence [53]. A modulation spectrum is used for advanced methods of GV in [54].

3.1.2 ARBM-based VC

An adaptive restricted Boltzmann machine (ARBM) [18] is proposed for VC. An ARBM-based VC requires neither the parallel data of a source speaker and target speaker, nor the parallel data of reference speakers.

3.1.2.1 Adaptive restricted Boltzmann machine

An ARBM is defined as a graphical and probabilistic model based on a restricted Boltzmann machine (RBM) [55]. RBM is an undirected graphical model that defines the distribution of visible unit with binary hidden units. In addition to these units, an ARBM has identity units that represent which speaker utters the sentence. Although an RBM was originally introduced as a method of representing binary-valued data, an ARBM uses real-valued data similar to Gaussian-Bernoulli RBM (GBRBM) [56, 57]. The joint probability $p(\boldsymbol{v}, \boldsymbol{h}, \boldsymbol{s})$ of real-valued visible units $\boldsymbol{v} \in \mathbb{R}^I$, binary-valued hidden units $\boldsymbol{h} \in \mathbb{B}^J$ ($\mathbb{B}$ is a space that takes 0 or 1.), and binary-valued identity units $\boldsymbol{s} \in \mathbb{B}^K$ is defined as follows:

$$
\begin{aligned}
p(\boldsymbol{v}, \boldsymbol{h}, \boldsymbol{s}) &= \frac{1}{Z} \mathrm{e}^{-E(\boldsymbol{v}, \boldsymbol{h}, \boldsymbol{s})}, \\
E(\boldsymbol{v}, \boldsymbol{h}, \boldsymbol{s}) &= \frac{1}{2}(\boldsymbol{v} - \boldsymbol{b})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{v} - \boldsymbol{b}) - \boldsymbol{v}^\top \boldsymbol{\Sigma}^{-1} \mathbf{W}(\boldsymbol{s}) \boldsymbol{h} - \boldsymbol{c}^\top \boldsymbol{h},
\end{aligned} \tag{3.26}
$$

where $\boldsymbol{b} \in \mathbb{R}^I$, $\boldsymbol{c} \in \mathbb{R}^J$, $\mathbf{W}(\boldsymbol{s}) \in \mathbb{R}^{I \times J}$, $\boldsymbol{\Sigma} = \mathrm{diag}(\boldsymbol{\sigma}^2)$ are model parameters of the ARBM, indicating a bias vector of visible units, a bias vector of hidden units,

the speaker-adaptive weight matrix between visible units and hidden units, a variance matrix of the visible units, respectively. I, J, and K indicate the number of visible units, hidden units, and identity units, respectively. The standard deviations associated with Gaussian visible units are $\boldsymbol{\sigma}^2 = [\sigma_1^2, \cdots, \sigma_I^2]$. $Z = \int^I \Sigma_{\boldsymbol{h},\boldsymbol{s}} \mathrm{e}^{-E(\boldsymbol{v},\boldsymbol{h},\boldsymbol{s})} d^I\boldsymbol{v}$ is the normalization constant.

A speaker-adaptive weight matrix is defined as follows:

$$\mathbf{W}(\boldsymbol{s}) = \mathcal{A} \otimes_3 \boldsymbol{s}\bar{\mathbf{W}} + \mathcal{B} \otimes_3 \boldsymbol{s}, \tag{3.27}$$

where $\bar{\mathbf{W}} \in \mathbb{R}^{I \times J}$ is a speaker-independent weight matrix, and third-order tensors $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$ and $\mathcal{B} \in \mathbb{R}^{I \times J \times K}$ adapt $\bar{\mathbf{W}}$ for the specific speaker. $\mathcal{X} \otimes_d \boldsymbol{y}$ indicates an operator that takes an inner product of a third-order tensor $\mathcal{X}$ unfolded along with the dth mode and a vector $\boldsymbol{y}$.

Because there are no connections between visible units and hidden units, the conditional probabilities form simple equations as follows:

$$p(\boldsymbol{v}|\boldsymbol{h}, \boldsymbol{s}) = \mathcal{N}(\boldsymbol{v}|\mathbf{W}(\boldsymbol{s})\boldsymbol{h} + \boldsymbol{b}, \boldsymbol{\Sigma}), \tag{3.28}$$

$$p(\boldsymbol{h}|\boldsymbol{v}, \boldsymbol{s}) = \mathcal{S}(\mathbf{W}(\boldsymbol{s})^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{v} + \boldsymbol{c}), \tag{3.29}$$

where $\mathcal{S}(\cdot)$ and $\mathcal{N}(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ indicate an element-wise sigmoid function and multivariate Gaussian probability density function with the mean $\boldsymbol{\mu}$, and variance-covariance $\boldsymbol{\Sigma}$.

Given a training data set $\{\boldsymbol{v}_n, \boldsymbol{s}_n\}_{n=1}^N$, the parameters are estimated to maximize the log-likelihood of a collection of visible units $\mathcal{L} = \log \prod_n p(\boldsymbol{v}_n, \boldsymbol{s}_n)$. Differentiating this log-likelihood with respective to θ, we obtain:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \langle \frac{\partial E(\boldsymbol{v}, \boldsymbol{h}, \boldsymbol{s})}{\partial \theta} \rangle_{\text{data}} - \langle \frac{\partial E(\boldsymbol{v}, \boldsymbol{h}, \boldsymbol{s})}{\partial \theta} \rangle_{\text{model}}, \tag{3.30}$$

where $\langle \cdot \rangle_{\text{data}}$ and $\langle \cdot \rangle_{\text{model}}$ indicate expectations of input data and the inner model, respectively. However, it is usually difficult to compute the second term in Eq. (3.30). Here, we use Contrastive Divergence [58] for the second term. Each parameter is updated using stochastic gradient descent (SGD) from Eq. (3.30).

3.1.3 Conversion based on ARBM

Each parameter of an ARBM is simultaneously estimated using training data that contains the speech uttered by some reference speakers. Then, using a small amount of speech data of the source speaker and the target speaker, we estimate the additional adaptive parameters while fixing speaker-independent parameters. Specifically, we replace $\mathcal{A}$ and $\mathcal{B}$ with $\mathbf{A}_s \cup_3 \mathbf{A}_t$ and $\mathbf{B}_s \cup_3 \mathbf{B}_t$, where $\mathbf{A}_s$, $\mathbf{A}_t$, $\mathbf{B}_s$, and $\mathbf{B}_t$ are an adaptive matrix for the source speaker, an adaptive matrix for the target speaker, a bias matrix for the source speaker, and a bias matrix for the target speaker, respectively. $\cup_d$ indicates a concatenate operation along with mode-d. In the conversion step, we calculate the latent features (hidden units) from the input features (acoustic features) of the source speaker $\boldsymbol{v}_s$ as follows:

$$\hat{\boldsymbol{h}} \triangleq \mathcal{S}((\mathbf{A}_s\bar{\mathbf{W}} + \mathbf{B}_s)^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{v}_s + \boldsymbol{c}). \tag{3.31}$$

Because the column vectors of the adapted weight matrix are similar to the patterns appearing in the source speaker's acoustic features, an ARBM assumes that the obtained latent features $\hat{\boldsymbol{h}}$ represent speaker-independent phonological information. Therefore, when we convert the speaker identity of the speech, we just calculate the visible units using the target identity without changing the phonological information. The converted acoustic features for the target speaker are obtained as follows:

$$\hat{\boldsymbol{v}}_t = (\mathbf{A}_t\bar{\mathbf{W}} + \mathbf{B}_t)\hat{\boldsymbol{h}} + \boldsymbol{b}. \tag{3.32}$$

This equation shows that the converted speech is generated from the phonological information $\hat{\boldsymbol{h}}$ and the target speaker-adaptive weight matrix.

3.2 Conventional ASR Methods

A typical ASR system consists of three components: signal processing, acoustic modeling, and language modeling. In the signal processing stage, a speech signal is converted into a sequence of an acoustic feature by the frame analysis. The major signal processing methods are described in Chapter 2.1. The acoustic modeling interprets these acoustic features into possible linguistic units, usually

words. The language modeling determines valid linguistic words or sentences. Among a process, it is not separable in practice. In the important progress of ASR, there are three techniques based on the DTW algorithm, hidden Markov models (HMM), and neural networks (NN).

3.2.1 DTW-based ASR

DTW, also known as dynamic programming (DP) matching, was introduced for non-linear time alignment of speech patterns. DTW can effectively minimize errors between two speech sequences. Suppose we are given the input acoustic pattern $X = \{x_1, ..., x_T\}$, the recognition class is estimated as follows:

$$\tilde{c} = \arg\min_{c} D(X, Y^c), \tag{3.33}$$

where $Y^c = \{y_1^c, ..., y_{T'}^c\}$ denotes the template that belongs the recognition class c. $D(\cdot, \cdot)$ is a distance measure between two inputs calculated by DTW.

The DTW-based approach is a non-parametric technique, and many speech templates are required to accommodate various uncertainties. This results in extensive computational load in the decoding procedure, as well as an extended training procedure. It has been shown that DTW can be considered as a special case of hidden Markov modeling, which is a parametric technique and offers flexibility and improved recognition accuracy [22, 23].

3.2.2 HMM-based ASR

In a HMM-based ASR framework, given an acoustic feature sequence $X = \{x_1, ..., x_T\}$, a word sequence $W = \{w_1, ..., w_I\}$ is estimated while maximizing the posterior probability $Pr(W|X)$ as follows:

$$\tilde{W} = \arg\max_{W} Pr(W|X) \tag{3.34}$$

$$= \arg\max_{W} Pr(X|W)Pr(W), \tag{3.35}$$

where $Pr(X)$ can be ignored because we optimize potential word sequences for the same observation X. The prior probability $Pr(W)$ of the word string itself is estimated by a language model. An acoustic model estimates the conditional

probability $Pr(X|W)$ that the generation probability of the acoustic feature sequence given the word sequence. Practically, in order to make it easier to handle, $Pr(X|W)$ is decomposed as follows:

$$Pr(X|W) \simeq \max_{P} Pr(X|P)Pr(P|W), \tag{3.36}$$

where $P = \{p_1, ..., p_M\}$ denotes the phoneme sequence. $Pr(P|W)$ is the pronunciation probability of the word, which is calculated from a pronunciation dictionary. Finally, the generation probability $Pr(X|P)$ can be written as follows:

$$Pr(X|P) \simeq \max_{S \in \Phi(P)} \prod_{t} Pr(x_t|s_t)Pr(s_t|s_{t-1}), \tag{3.37}$$

where $S = \{s_1, ..., s_T\}$ and $\Phi(P)$ denote the hidden state sequence of HMMs and a set of the hidden state sequence generated from the phoneme sequence P, respectively. On ASR, a typical HMM represents output distributions by Gaussian mixture model (GMM), then it is called GMM-HMM. In GMM-HMM systems, $Pr(x_t|s_t)$ is defined as follows:

$$Pr(x_t|s_t) = \sum_{k} w_{s_t,k} \mathcal{N}(x_t; \mu_{s_t,k}, \Sigma_{s_t,k}), \tag{3.38}$$

where $\mu_{s_t,k}$, $\Sigma_{s_t,k}$ and $w_{s_t,k}$ are the mean vector and the variance matrix in state s_t with the kth mixture, and the mixture weight, respectively.

3.2.3 NN-based ASR

In the area of speech processing, the advent of deep learning has given rise to a renewed interest in NN models. The current main stream of ASR is based on a hybrid deep neural networks (DNN)-HMM [25] where DNN are used to calculate the output probability $Pr(x_t|s_t)$ of HMMs. Hybrid DNN-HMM systems give a significant improvement over GMM-HMM ASR systems.

In the DNN-HMM ASR procedure, we first train the acoustic model based on GMM-HMM, then calculate a frame-wise state alignment. Next, we train DNNs that predict the hidden state from the acoustic feature using the obtained alignment as a correct label. If the jth kind of the hidden state is correct one, the output of DNNs is enforced to be 1 at the jth dimension as well as 0 at the

other dimensions. The output of DNNs represents the posterior probability for hidden states as follows:

$$y_{t,j} = Pr(s_t = j|x_t). \tag{3.39}$$

Because the HMM procedure requires the output probability $Pr(x_t|s_t)$, this is replaced with the posterior probability $Pr(s_t|x_t)$ by using so-called pseudo-likelihood trick [25] as follows:

$$Pr(x_t|s_t) = \frac{Pr(s_t|x_t)Pr(x_t)}{Pr(s_t)} \tag{3.40}$$

$$\propto \frac{Pr(s_t|x_t)}{Pr(s_t)}, \tag{3.41}$$

where $Pr(x_t)$ is ignored because it does not depend on the state s_t. $Pr(s_t)$ can be regarded as the relative frequency of the state s_t in the training data.

The powerful DNN feature extraction can be one of the important properties. GMM-HMM optimizes the Gaussian distribution from the handcrafted feature. In contrast, DNNs that have multiple nonlinear transformations that projects the input feature into an optimal space for discrimination and distinguishes the state in that space. Therefore, the optimized feature extractor is incorporated into DNN-HMMs, which is one of the reasons for performance improvement.

More recently, end-to-end (E2E) ASR [28, 59] has become a popular alternative to greatly simplify the model-building process of conventional ASR systems by representing complicated modules with a single NN architecture. E2E ASR models $Pr(W|X)$ using DNNs. There are two major types of E2E architectures: connectionist temporal classification (CTC) and attention-based method. The CTC [60] uses Markov assumptions to efficiently solve sequential problems by dynamic programming. The attention-based method [61] that has an encoder and decoder modules uses an attention mechanism to perform alignment between acoustic frames and linguistic units.

Chapter 4

Non-parallel dictionary learning for voice conversion using non-negative Tucker decomposition

The related publications for this chapter are [62].

4.1 The Motivation and Related Work

Parallel data — aligned speech data from the source and the target speakers, so that each frame of the source speaker's data corresponds to that of the target speaker's data — leads to several problems in the task of voice conversion (VC). First, the data are limited to predefined statements (both speakers must utter the same statements). Second, the training data (the parallel data) are not the original speech data anymore, as the speech data are stretched and modified along the time axis when aligned, and there is no certainty that each frame is aligned perfectly. In the case of persons with articulation disorders, it is especially difficult to collect their speech for calculating the parallel data owing to athetoid symptoms. Hence, in this chapter, we propose a nonparallel VC method.

VC is a technique used to convert speaker-specific information in the speech of a source speaker into that of a target speaker while retaining linguistic in-

formation. Lately, VC techniques have been garnering particular attention [63], and various statistical approaches to VC have been studied [8, 9] as these techniques can be applied to numerous tasks [64, 65, 66, 67, 68]. Of these approaches, the Gaussian mixture model (GMM)-based mapping method [10] is the most prevalent, and a number of enhancements have been proposed [11, 12, 13]. Other VC methods, such as approaches based on non-negative matrix factorization (NMF) [14, 15, 69], neural networks [16], deep learning [70, 71], restricted Boltzmann machines [17, 18, 72], variational autoencoders [73], and a generative adversarial network [74], have also been proposed. Notably, in recent years, the NMF has outperformed GMM in parallel data conditions. Exemplar-based NMF-VC retains the high naturality of the converted speech, and many of its variants have been proposed [75, 76]. Although more recent deep learning methods require significantly large training data, NMF-VC requires comparatively less training data. Therefore, this study focuses on NMF-VC.

NMF [20] is one of the most popular sparse representation methods. The goal of NMF is to decompose the input observation into two matrices: the basis matrix and weight matrix. In this study, the basis matrix is referred to as the "dictionary," and the weight matrix as the "activity." The NMF-based method can be classified into two approaches: the dictionary-learning approach [15] and exemplar-based approach [77]. In the dictionary-learning approach, the dictionary and activity are estimated simultaneously during the training, and the estimated dictionary is used in conversion. However, in the exemplar-based approach, the training data is straightaway used as exemplars in the conversion step. By using the learned dictionary instead of the exemplars, the VC is executed with lower computation times.

However, both the NMF-based approaches require parallel data for training the models. As the dictionary is assembled from parallel data, the error of alignment in the parallel data might adversely affect VC performance. Several other approaches have been proposed that do not use (or minimally use) parallel data of the source and the target speakers [32, 78, 79]. For example, in [78], the spectral relationships between two arbitrary speakers (reference speakers) is modeled using GMMs and the source speaker's speech is converted using the matrix that projects the feature space of the source speaker into that of the target speaker

through that of the reference speakers. In this study, the conventional NMF-based VC method is expanded into a nonparallel VC method. A previous study [32] proposed using the phone segmentation results from automatic speech recognition to construct a sub-dictionary for each phone for an exemplar-based NMF voice conversion. This particular technique was applied to the nonparallel VC.

To tackle the nonparallel approach, a non-negative Tucker decomposition (NTD) [80, 81, 82]-based dictionary-learning method is proposed. The NTD is a non-negative extension of the Tucker decomposition that decomposes the input observation into a set of matrices and one core tensor. Tucker decomposition is generally introduced to deal with a high-order tensor. In recent studies, Tucker decomposition has been widely applied in visual question-answering systems [83] and speech recognition [84]. As spectral features are used for input observation, a set of matrices consists of two mode matrices for frequency and time and a core tensor corresponding to a core matrix. It is assumed that these matrices correspond to the frequency basis matrix, the phonemic information, and a codebook between the frequency basis and each phone, respectively. In the proposed approach, the activity matrix in NMF is decomposed into the codebook and the phonemic information. When learning the dictionaries, while the activity matrix is shared between speakers using parallel data in the conventional NMF-VC, in the proposed method, the codebook is shared between speakers and the phonemic information is dependent on a speaker. Hence, the time-varying phonemic information can be captured for each speaker. During the conversion, only the phonemic information matrix is estimated as the activity matrix. As the proposed method can have time-dependent factors for each speaker, there is no necessity for parallel data. To the best of authors' knowledge, NTD-based VC has not been attempted, except [85] where Tucker decomposition was used to represent the speaker space and the conversion mechanism was based on GMM. The present VC is based on NMF, and this approach is fundamentally different from those presented previously [85]. Several methods have been proposed for tensor decomposition [86, 87, 88]. In [86], NMF is applied to variational Bayesian matrix factorization, where each observed entry is assumed to be a beta distribution. Shi *et al.* [87] proposed tensor decomposition with variance maximization for feature extraction. In [88], pairwise similarity information is incorporated into Tucker

tensor decomposition. While these methods have useful properties, it is difficult to adapt them directly to VC. NTD can be readily integrated with NMF-based VC, because NMF is the 2nd-order case of the Tucker decomposition with the non-negative constraint.

4.2 NMF-based Voice Conversion

NMF is a matrix decomposition method under non-negative constraints. The basic idea behind decomposing a matrix $\mathbf{X} \in \mathbb{R}^{F \times T}$ is to find two matrices $\mathbf{W} \in \mathbb{R}^{F \times K}$ and $\mathbf{H} \in \mathbb{R}^{K \times T}$ that minimize the distance between $\mathbf{X}$ and $\mathbf{WH}$ under non-negative constraints. F and T represent the number of dimensions and frames. In NMF, $\mathbf{W}$ is called a basis matrix and contains K bases in columns. $\mathbf{H}$ is called an activity matrix and indicates the activity of each basis along the time index.

VC approaches using NMF are divided into two categories: supervised and unsupervised approaches. The supervised approach, known as the exemplar-based VC, estimates only the activity from observation and the dictionary must be provided. However, the unsupervised approach, i.e., the dictionary-learning VC, estimates both the dictionary and the activity from observation. The proposed method is based on the latter, i.e., the dictionary learning approach.

4.2.1 Dictionary learning using NMF

Fig. 4.1 shows the basic approach of the dictionary-learning NMF-based VC [15], where F, T, and K represent the number of dimensions, frames, and bases, respectively. This VC method needs two dictionaries that are phonemically parallel. $\mathbf{W}^s \in \mathbb{R}^{F \times K}$ represents a source dictionary, and $\mathbf{W}^t \in \mathbb{R}^{F \times K}$ represents a target dictionary. In exemplar-based VC, these two dictionaries consist of the same words or sentences and are aligned with dynamic time warping (DTW), which is comparable with the conventional GMM-based VC. In dictionary-learning VC, these two dictionaries are estimated simultaneously and as a result have the same number of bases.

For the training source speaker data $\mathbf{X}^s \in \mathbb{R}^{F \times T}$ and the training target speaker data $\mathbf{X}^t \in \mathbb{R}^{F \times T}$, two dictionaries $\mathbf{W}^s$, $\mathbf{W}^t$, and the activity $\mathbf{H} \in \mathbb{R}^{K \times T}$

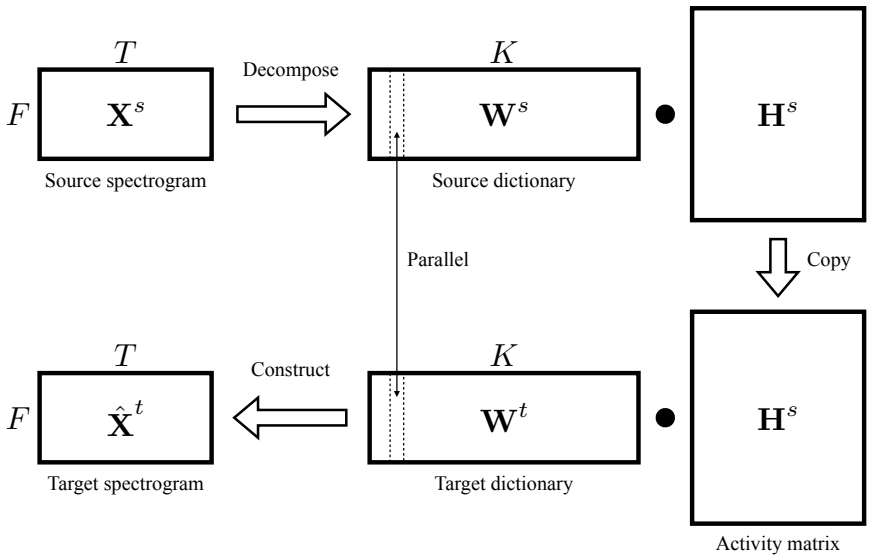


Figure 4.1: Basic approach of NMF-based voice conversion.

are simultaneously estimated. The cost function of this joint NMF is defined as follows:

$$d_{KL}(\mathbf{X}^s, \mathbf{W}^s\mathbf{H}) + d_{KL}(\mathbf{X}^t, \mathbf{W}^t\mathbf{H}) + \lambda||\mathbf{H}||_1$$
$$s.t.\ \forall i, j\ \mathbf{W}^s_{ij}, \mathbf{W}^t_{ij}, \mathbf{H}_{ij} \geq 0, \tag{4.1}$$

where $\mathbf{X}^s$ and $\mathbf{X}^t$ represent parallel data, and $\forall i, j\ \mathbf{A} \geq 0$ indicates that each element of a matrix $\mathbf{A}$ has a non-negative value. In Eq. (4.1), $d_{KL}(\mathbf{A}, \mathbf{B})$ denotes the Kullback-Leibler divergence between the two matrices $\mathbf{A}$ and $\mathbf{B}$, and the last term is the sparsity constraint with the L1-norm regularization term that causes the activity matrix to be sparse. λ represents the weight of the sparsity constraint. This function is minimized by iteratively updating parameters, as is done in the traditional NMF.

This method assumes that when the source and the target spectra (which are from the same words but spoken by different speakers) are expressed with sparse representations of the source dictionary and the target dictionary, respectively, the obtained activity matrices are approximately equivalent to each other. In the conversion process, for the input source spectrogram $\mathbf{X}^s$, only the activity $\mathbf{H}^s$ is estimated while fixing the source dictionary $\mathbf{W}^s$. The estimated source activity

$\mathbf{H}^s$ is multiplied with the target dictionary $\mathbf{W}^t$, and the target spectrogram $\hat{\mathbf{X}}^t$ is constructed as follows:

$$\hat{\mathbf{X}}^t = \mathbf{W}^t\mathbf{H}^s. \tag{4.2}$$

4.2.2 Problems

NMF-based VC has several problems. First, if the source and target utterances are aligned using DTW in advance, the estimated parameters are affected by the quality of the alignment. And a mismatch of alignment appears to persist. Aihara *et al.* [75] have shown that this mismatch degrades the performance of exemplar-based VC. Second, it appears that the activity matrix contains other information along with the phonetic information. Aihara *et al.* [76, 77] assumed that the activity matrix contains the phonetic information and speaker information, and accordingly proposed certain frameworks to overcome this effect, thereby improving the performance of NMF-based VC. In this study, an alternative approach is proposed. The activity matrix is decomposed into the speaker-shared matrix and the speaker-dependent phonetic information matrix. This decomposition makes parallel data unnecessary. Moreover, during the conversion, estimating only the phonetic information matrix as the activity matrix is expected to improve the accuracy of activity estimation.

4.3 Proposed Method

4.3.1 NTD

Given a non-negative N-way tensor, NTD [89] decomposes the input tensor into a core tensor and a set of mode matrices that are restricted to have only non-negative elements. In this study, as the spectral features are used as the input observation, a core tensor is represented as a matrix, and there are two mode matrices. Under these conditions, NTD is simply defined as follows:

$$\mathbf{X} \approx \mathbf{U}\mathbf{G}\mathbf{V}^\top \ s.t.\ \forall i, j\ \mathbf{U}_{ij}, \mathbf{G}_{ij}, \mathbf{V}_{ij} \geq 0, \tag{4.3}$$

where $\mathbf{X} \in \mathbb{R}^{F \times T}$, $\mathbf{U} \in \mathbb{R}^{F \times M}$, $\mathbf{V} \in \mathbb{R}^{T \times L}$, and $\mathbf{G} \in \mathbb{R}^{M \times L}$ represent an input spectrogram, a mode matrix along the frequency axis, a mode matrix along the time axis, and a core matrix, respectively. F, T, M, and L indicate the number of frequency bins, frames, frequency basis, and time basis, respectively. The cost function of NTD is defined as follows:

$$d_{KL}(\mathbf{X}, \mathbf{U}\mathbf{G}\mathbf{V}^{\top}). \tag{4.4}$$

NTD provides a general form of the non-negative tensor factorization including a special case of NMF; updating algorithms have been proposed in [89]. These updating algorithms are based on that NMF.

4.3.2 Dictionary learning using NTD

This section describes the method of estimating a parallel dictionary between the source and target speakers by NTD. The objective function is represented as follows:

$$\begin{aligned} &\alpha d_{KL}(\mathbf{X}^s, \mathbf{U}^s\mathbf{G}\mathbf{V}^{s\top}) + \beta d_{KL}(\mathbf{X}^t, \mathbf{U}^t\mathbf{G}\mathbf{V}^{t\top}) \\ &\qquad + \lambda||\mathbf{V}^{s\top}||_1 + \lambda||\mathbf{V}^{t\top}||_1 \\ &s.t.\ \forall i, j\ \mathbf{U}^s_{ij}, \mathbf{U}^t_{ij}, \mathbf{G}_{ij}, \mathbf{V}^s_{ij}, \mathbf{V}^t_{ij} \geq 0, \end{aligned} \tag{4.5}$$

where $\mathbf{X}^s \in \mathbb{R}^{F \times T_s}$, $\mathbf{X}^t \in \mathbb{R}^{F \times T_t}$, $\mathbf{U}^s \in \mathbb{R}^{F \times M}$, $\mathbf{U}^t \in \mathbb{R}^{F \times M}$, $\mathbf{V}^s \in \mathbb{R}^{T_s \times L}$, $\mathbf{V}^t \in \mathbb{R}^{T_t \times L}$, and $\mathbf{G} \in \mathbb{R}^{M \times L}$ represent the source and target spectrograms, the source and target frequency basis matrices, the source and target time basis matrices, and a core matrix, respectively. α and β represent the weight of each term. F, T_s, T_t, M, and L indicate the number of frequency bins, source and target frames, frequency basis, and time basis, respectively. This function is minimized

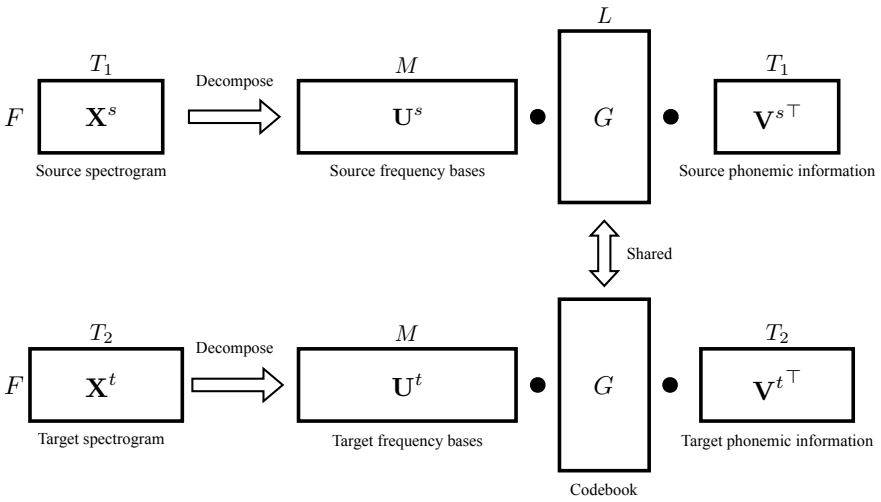


Figure 4.2: Decomposition of spectrograms to frequency bases and phonemic information, using NTD.

by iteratively updating the following equations in the same manner as the NTD:

$$\mathbf{U}^s \leftarrow \mathbf{U}^s .* (\widetilde{\mathbf{H}}^s (\mathbf{X}^s ./ \mathbf{U}^s \widetilde{\mathbf{H}}^s)^\top ./ \widetilde{\mathbf{H}}^s \mathbf{1}^{(T_s \times F)})^\top \tag{4.6}$$

$$\mathbf{U}^t \leftarrow \mathbf{U}^t .* (\widetilde{\mathbf{H}}^t (\mathbf{X}^t ./ \mathbf{U}^t \widetilde{\mathbf{H}}^t)^\top ./ \widetilde{\mathbf{H}}^t \mathbf{1}^{(T_t \times F)})^\top \tag{4.7}$$

$$\mathbf{V}^s \leftarrow \mathbf{V}^s .* ((\mathbf{X}^s ./ \widetilde{\mathbf{W}}^s \mathbf{V}^{s\top})^\top \widetilde{\mathbf{W}}^s) ./ (\mathbf{1}^{(T_s \times F)} \widetilde{\mathbf{W}}^s + \lambda \mathbf{1}^{(T_s \times L)}) \tag{4.8}$$

$$\mathbf{V}^t \leftarrow \mathbf{V}^t .* ((\mathbf{X}^t ./ \widetilde{\mathbf{W}}^t \mathbf{V}^{t\top})^\top \widetilde{\mathbf{W}}^t) ./ (\mathbf{1}^{(T_t \times F)} \widetilde{\mathbf{W}}^t + \lambda \mathbf{1}^{(T_t \times L)}) \tag{4.9}$$

$$\begin{aligned} \mathbf{G} \leftarrow & \mathbf{G} .* (\mathbf{U}^{s\top} (\mathbf{X}^s ./ \widetilde{\mathbf{X}}^s) \mathbf{V}^s + \mathbf{U}^{t\top} (\mathbf{X}^t ./ \widetilde{\mathbf{X}}^t) \mathbf{V}^t) \\ & ./ (\mathbf{U}^{s\top} \mathbf{1}^{(F \times T_s)} \mathbf{V}^s + \mathbf{U}^{t\top} \mathbf{1}^{(F \times T_t)} \mathbf{V}^t) \end{aligned} \tag{4.10}$$

$$\begin{aligned} \widetilde{\mathbf{H}}^s = & \mathbf{G} \mathbf{V}^{s\top}, \widetilde{\mathbf{H}}^t = \mathbf{G} \mathbf{V}^{t\top}, \widetilde{\mathbf{W}}^s = \mathbf{U}^s \mathbf{G}, \widetilde{\mathbf{W}}^t = \mathbf{U}^t \mathbf{G}, \\ \widetilde{\mathbf{X}}^s = & \mathbf{U}^s \mathbf{G} \mathbf{V}^{s\top}, \widetilde{\mathbf{X}}^t = \mathbf{U}^t \mathbf{G} \mathbf{V}^{t\top}, \end{aligned}$$

where $.*$ and $./$ denote elementwise multiplication and division, respectively. In this framework, only a core matrix $\mathbf{G}$ is shared, and time-varying matrices $\mathbf{V}^s$ and $\mathbf{V}^t$ are dependent on each speaker, as shown in Fig. 4.2. Therefore, there is no necessity for parallel data.

After each matrix in the model is estimated, the source and target parallel dictionaries are calculated as $\mathbf{U}^s\mathbf{G}$ and $\mathbf{U}^t\mathbf{G}$, respectively. During conversion, for the given source spectrogram $\mathbf{X}^s$, only $\mathbf{V}^s$ is estimated as $\mathbf{X}^s = \mathbf{U}^s\mathbf{G}\mathbf{V}^{s\top}$. Then, the target spectrogram $\hat{\mathbf{X}}^t$ is obtained as $\hat{\mathbf{X}}^t = \mathbf{U}^t\mathbf{G}\mathbf{V}^{s\top}$.

It is assumed that $\mathbf{U}^s$ and $\mathbf{U}^t$ represent the frequency basis matrices, and $\mathbf{V}^s$ and $\mathbf{V}^t$ represent the phonemic information. As the core matrix is not dependent on either the frequency or the time, this matrix represents the codebook between the frequency bases and the phones. Based on this assumption, the core matrix makes a correspondence between frequency bases and phones. Specifically, there are L phones, and a spectrum of each phone is constructed using M frequency bases. Although the information contained in the activity matrix is not only the phonemic information, in conventional NMF-based approaches, the activity matrix is assumed to contain only the phonemic information. Therefore, the estimated activity is degraded. In contrast, the proposed NTD-based approach specifically decomposes the activity matrix into the speaker-shared information and the speaker-dependent phonemic information. Therefore, it is expected that the performance of the activity estimation will be improved during conversion.

4.4 Experiments

4.4.1 Experimental Conditions

The proposed VC technique was evaluated in a speaker-conversion task using clean speech data by comparing its results with the conventional GMM-based method [11], the conventional NMF-based dictionary-learning method [15], and an adaptive restricted Boltzmann machine (ARBM)-based method [18] that does not use parallel data. For the evaluation, voice samples of speech data stored in the ATR Japanese speech database [90] of three males and three females were used. The sampling rate was 16 kHz. A total of 45 sentences were used for training, and another 50 sentences were used for testing. Parallel data aligned using dynamic programming matching (DPM) was used to train the GMM-based and NMF-based methods. The proposed method and the ARBM-based method do not require parallel data. As training data, the same utterances were used for the source and the target speaker in the parallel setting, and completely different utterances for each speaker were used in the nonparallel setting.

Parameter initialization has a significant impact on the conversion performance. In this study, $\mathbf{V}^s$ and $\mathbf{V}^t$ are initialized randomly. Table 4.1 shows the

Table 4.1: Algorithm for initializing parameters

Initializing in the parallel setting
• Set source and target parallel data $\mathbf{X}_s$ and $\mathbf{X}_t$
• Optimize $\mathbf{W}_s$, $\mathbf{W}_t$, and $\mathbf{H}$ minimizing $d_{KL}(\mathbf{X}_s, \mathbf{W}_s\mathbf{H}) + d_{KL}(\mathbf{X}_t, \mathbf{W}_t\mathbf{H})$
• Optimize $\mathbf{U}_s$, $\mathbf{U}_t$, and $\mathbf{G}$ minimizing $d_{KL}(\mathbf{W}_s, \mathbf{U}_s\mathbf{G}) + d_{KL}(\mathbf{W}_t, \mathbf{U}_t\mathbf{G})$
Initializing in the nonparallel setting
• Set source training data $\mathbf{X}_s$
• Optimize $\mathbf{W}_s$ and $\mathbf{H}_s$ while minimizing $d_{KL}(\mathbf{X}_s, \mathbf{W}_s\mathbf{H}_s)$
• Set target training data $\mathbf{X}_t$
• Optimize $\mathbf{A}$ and $\mathbf{H}_t$ while minimizing $d_{KL}(\mathbf{X}_t, \mathbf{A}\mathbf{W}_s\mathbf{H}_t)$ while fixing $\mathbf{W}_s$
• Optimize $\mathbf{U}_s$, $\mathbf{U}_t$, and $\mathbf{G}$ while minimizing $d_{KL}(\mathbf{W}_s, \mathbf{U}_s\mathbf{G}) + d_{KL}(\mathbf{A}\mathbf{W}_s, \mathbf{U}_t\mathbf{G})$

initialization algorithm for $\mathbf{U}^s$, $\mathbf{U}^t$ and $\mathbf{G}$. In the parallel setting, the initialization is based on the NMF framework using parallel data calculated by the source and target training data. In the nonparallel setting, the initialization is based on the NMF and NTD frameworks. This initialization method uses an adaptive matrix [91]. Finally, initialized parameters are optimized by Equations (4.6) to (4.10).

In the conventional NMF-based method and the proposed method, a 513-dimensional WORLD spectrum [41] is used for spectral features. The hyperparameters α and β are used to control the length of the training data for the source and the target speaker, respectively. These parameters were set as follows:

$$\alpha = min(T_s, T_t)/T_s \tag{4.11}$$

$$\beta = min(T_s, T_t)/T_t, \tag{4.12}$$

where T_s and T_t represent the number of frames of source and target training data, respectively. The sparse constraint λ was set to 0.2. The parameters are updated until the convergence condition $|F_t - F_{t-1}| < \epsilon|F_T|$ is fulfilled, where $|F_t|$ indicates a value of an objective function at an iteration t. ϵ was set to $\exp(-9)$. The GMM experiments were implemented using sprocket [92]. In the conventional NMF-based dictionary-learning method, the number of bases is 1,000. In the ARBM-based method, a 32-dimensional Mel-cepstrum that was calculated from

the 513-dimensional WORLD spectra was used as an input vector. Softmax constraints were set to hidden units.

In this study, a conventional linear regression based on the mean and standard deviation [11] was used to convert F0 information. Other information, such as aperiodic components, was synthesized without any conversion.

The proposed method was evaluated both objectively and subjectively. Mel-cepstral distortion (MCD) [dB] was used as a measure of the objective evaluations, defined as follows:

$$MCD = (10/\ln 10)\sqrt{2\sum_{d=1}^{24}(mc_d^{conv} - mc_d^{tar})^2} \tag{4.13}$$

where mc_d^{conv} and mc_d^{tar} represent the d-th dimension of the converted and target Mel-cepstral coefficients, respectively.

The subjective evaluation was based on "speech quality" and "similarity to the target speaker (individuality)". In the subjective evaluation, 25 sentences were evaluated by 10 native Japanese speakers. To evaluate the speech quality, a mean opinion score (MOS) test was performed. The opinion score was set to a five-point scale (5: excellent, 4: good, 3: fair, 2: poor, 1: bad). For the similarity evaluation, an XAB test was conducted, in which each participant listened to the voice of the target speaker and then to the voice converted using the two methods. The participant was then asked to judge which sample sounded most similar to the target speaker's voice.

4.4.2 Parameters

The performance of each method was evaluated using different parameters. One male speaker and one female speaker were selected and male-to-female conversion and female-to-male conversion was evaluated.

First, the performance of the conventional GMM-based VC was evaluated using different number of mixtures. The results obtained when using 4, 8, 16, 32, 64, and 128 mixtures are shown in Fig. 4.3. A lower value indicates a better result. As shown in Fig. 4.3, the optimal numbers were close to 8. Therefore, eight mixtures were used in the subsequent experiments.

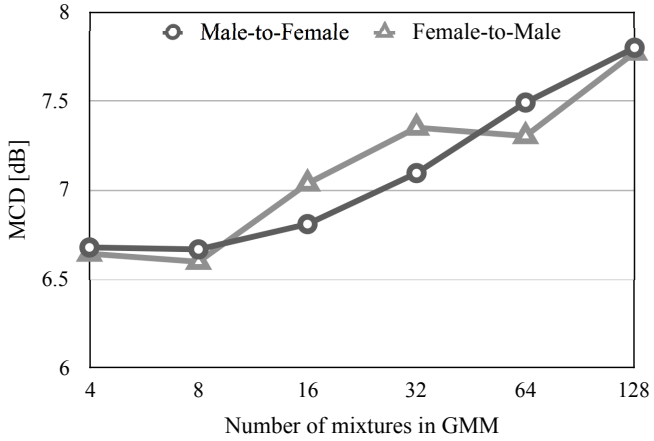


Figure 4.3: Average MCD [dB] of the conventional GMM-based VC when varying the number of mixtures.

Next, the performance of the conventional ARBM-based VC was evaluated using a different number of hidden units. The results are shown in Fig. 4.4 when using 2, 4, 8, 16, 32, and 48 hidden units. As shown in Fig. 4.4, the optimal number was around 32. Therefore, 32 hidden units were used in the later experiments.

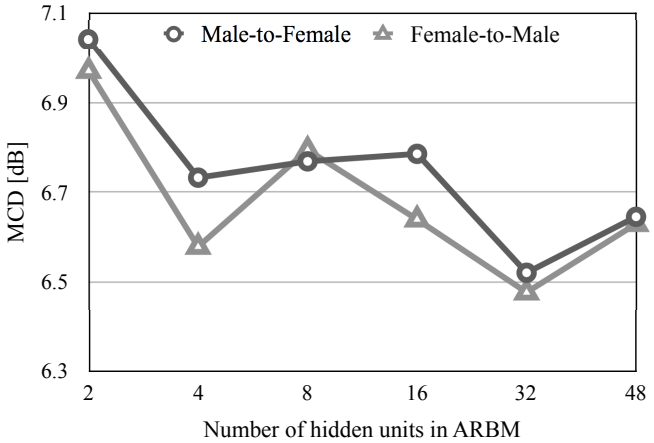


Figure 4.4: Average MCD [dB] of the conventional ARBM-based VC when varying the number of hidden units.

Finally, the performance of the proposed method was evaluated using a different number of frequency bases. The results are shown in Fig. 4.5 when the

numbers of frequency bases M were 100, 200, 300, 400, and 500. The optimal number was around 200. Therefore, 200 was used as the number of frequency bases in the subsequent experiments. In the experiments, to control the number of dictionary bases during conversion, the number of time bases L was fixed to 1,000.

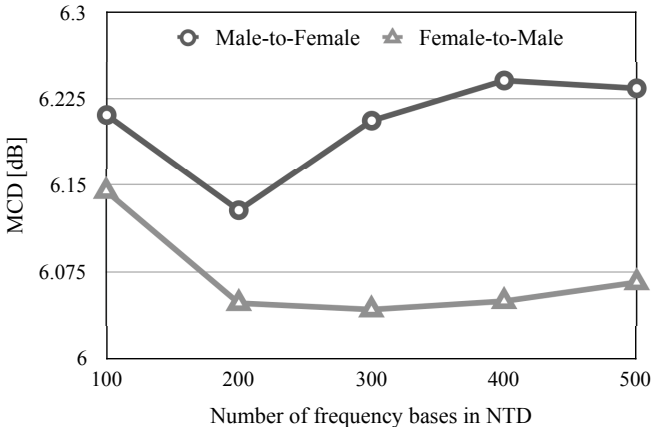


Figure 4.5: Average MCD [dB] of the conventional NTD-based VC when varying the number of frequency bases.

4.4.3 Experimental Results

In this section, the proposed method is compared with conventional GMM, NMF, and ARBM-based methods.

Initially, the proposed method is compared with the parallel method in a parallel setting. Table 5.3 shows the average MCD values for male-to-female conversion, female-to-male conversion, male-to-male conversion, and female-to-female conversion. In this table, "Mi" and "Fj" indicate the i-th male speaker and j-th female speaker, respectively, and $src \rightarrow tar$ denotes the src-to-tar conversion. The rightmost column in the table indicates the mean value for each method with a 95% confidence interval. Here, a lower value indicates a better result. In these experiments, the models were trained using parallel utterances. The GMM and NMF frameworks require parallel data. For these, parallel utterances were used to calculate the parallel data. Table 5.3 clearly demonstrates that the proposed

Table 4.2: Average MCD [dB] using parallel utterances

	Male-to-Female			Female-to-Male			Male-to-Male		Female-to-Female		mean
	M1→F1	M2→F2	M3→F3	F1→M1	F2→M2	F3→M3	M1→M3	M3→M2	F1→F3	F3→F2	
GMM	6.67	7.35	6.76	6.60	6.97	7.04	6.41	7.36	6.42	7.21	6.88 ± 0.04
NMF	6.24	6.62	6.32	6.14	6.34	6.23	6.20	6.68	6.11	6.08	6.30 ± 0.03
NTD	6.12	6.50	6.31	6.04	6.23	6.08	6.05	6.66	5.99	5.86	6.19 ± 0.03

Table 4.3: Average MCD [dB] using nonparallel data

	Male-to-Female			Female-to-Male			Male-to-Male		Female-to-Female		mean
	M1→F1	M2→F2	M3→F3	F1→M1	F2→M2	F3→M3	M1→M3	M3→M2	F1→F3	F3→F2	
ARBM	6.52	6.69	6.27	6.48	6.76	6.62	6.98	7.04	6.37	6.21	6.59 ± 0.03
NTD	6.75	7.30	6.56	6.75	7.04	6.99	6.85	7.04	6.64	6.03	6.67 ± 0.04

NTD-based dictionary learning is not affected by the alignment error in DTW, and hence yields 10.1% and 1.8% relative improvements when compared with the conventional GMM-based method and the conventional NMF-based dictionary learning, respectively. Moreover, it confirms that the proposed method achieved a significantly better score than both the comparative methods, when using a p-value test of 0.05.

Next, the method was compared with the nonparallel method in a nonparallel setting. Table 4.3 shows the average MCD values for male-to-female conversion, female-to-male conversion, male-to-male conversion, and female-to-female conversion. These results show that the proposed method has a comparable performance to the conventional nonparallel method, ARBM. However, the proposed method achieved a notably worse score than the ARBM-based method, when using a p-value test of 0.05. This difference is explained in the next section.

Fig. 4.6 shows the results of the MOS test on speech quality. The error bar shows a 95% confidence interval. Here, a higher value indicates a better result. M-to-F, F-to-M, M-to-M, and F-to-F denote male-to-female conversion, female-to-male conversion, male-to-male conversion, and female-to-female conversion, respectively. "NTD (para)" and "NTD (non-para)" denote the proposed method with parallel utterances training and nonparallel utterances training, respectively. The proposed method achieved a significantly better score than the conventional methods. Specifically, NTD with the nonparallel setting showed the best results across all conversions.

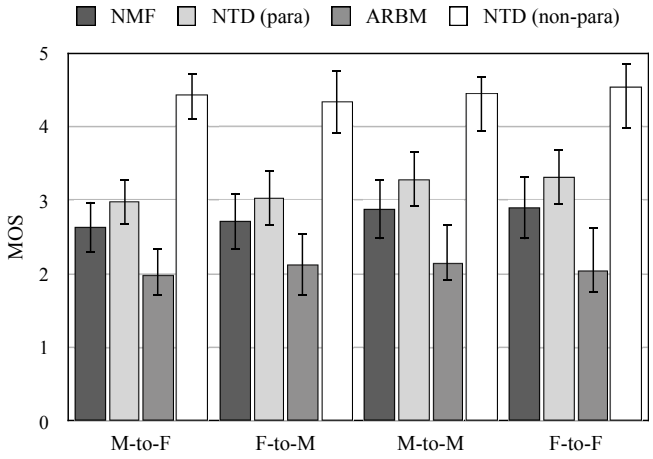


Figure 4.6: MOS of speech quality.

Fig. 4.7 and Fig. 4.8 show the results of the XAB test. The error bar shows a 95% confidence interval. For this test, a higher value indicates a better result. In Fig. 4.7, the results of the proposed method and conventional NMF-based dictionary-learning method are compared. In the male-to-female and female-to-female conversions, the proposed method achieved a better score than NMF-based dictionary learning. In the male-to-male and female-to-male conversions, the proposed method achieved a lower score than NMF-based dictionary learning. However, the difference between the two methods is not statistically significantly, because $p > 0.3$ in the p-value test. The proposed NTD-based dictionary learning without calculating parallel data showed comparable performance to the conventional NMF-based dictionary learning, which requires parallel data. In Fig. 4.8, the results of the proposed method and the ARBM-based VC are compared. In conversions to male, the proposed method achieved a better score than ARBM-base VC. In conversions to female, the proposed method achieved a lower score than ARBM-based VC. In only the male-to-female conversion, the difference was significant — $p < 0.05$. However, in other conversions, the difference was not statistically significant. These tests show that the proposed nonparallel VC approach effectively converts the individuality of the source speaker's voice to the target speaker's voice while preserving high speech quality.

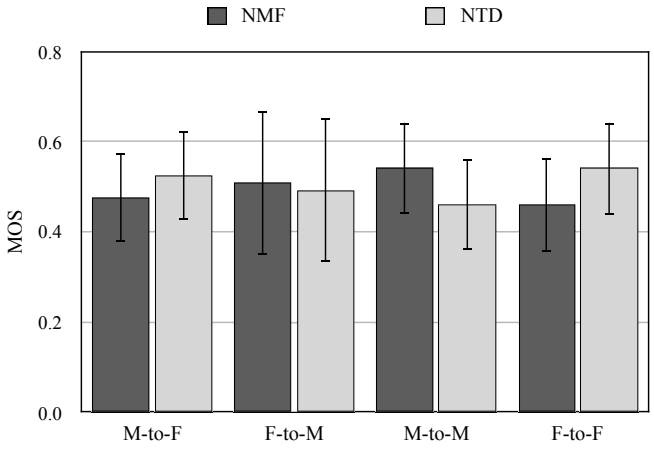


Figure 4.7: XAB test between NMF and NTD.

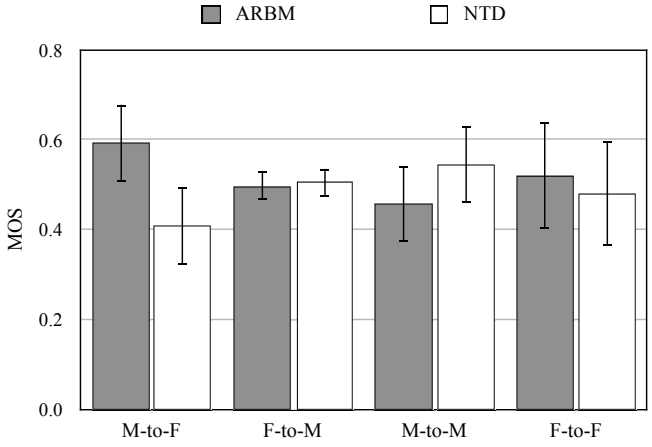


Figure 4.8: XAB test between ARBM and NTD.

4.4.4 Discussion

In the objective evaluations, the proposed method achieved a better MCD value than the conventional VC, which uses parallel data. This is due to the fact that the proposed method is not affected by the mismatch of DPM. Moreover, the proposed NTD-based method yielded better performance, although the number of learned parameters decreased by approximately 60% of the conventional NMF-based one. This result indicates that the proposed dictionary learning has better spectral representation while keeping the number of bases of dictionaries constant during conversion. In addition, the average difference in MCD between the pro-

posed method and the ARBM-based method was approximately 0.08 dB. This difference is relatively small. It is assumed that MCD is superior to the ARBM-based method, as it uses Mel-cepstrum as an input feature, whereas NTD-based methods use a WORLD spectrum. In the speech quality test, the proposed method using nonparallel training data achieved a better MOS score than that using parallel utterances. This is due to the model's ability to learn diverse phonemic information by using nonparallel data when compared with parallel utterances. For example, n sentences are used for each speaker as training data. In the instances using parallel utterances, which consist of the same context for both speakers, the frequency base matrices $\mathbf{U}^s$ and $\mathbf{U}^t$ and the codebook $\mathbf{G}$ are learned from n context patterns. However, in the nonparallel setting, where a different context was used for the source and target speakers, the frequency base matrices and the codebook were learned from n and $2n$ context patterns, respectively. A codebook was effectively learned while improving the generalization ability. Therefore, the method using nonparallel data outperformed that using parallel utterances.

4.4.5 Experiments on Voice Conversion Challenge 2018

The proposed method was also evaluated on the Voice Conversion Challenge (VCC) 2018 [93], which includes both parallel and nonparallel recordings from native English speakers from the US. VCC 2018 consists of a total of 12 speakers. Each speaker has sets of 81 and 35 sentences for training and evaluation, respectively. The recordings were down sampled to 16 kHz. Systems were conducted for the 16 combinations of source-target pairs.

The results of this objective evaluation are shown in Table 4.4. Our proposed method did not outperform the GMM-based VC in the parallel setting, while the NTD-based method achieved 3.89% relative improvement compared with the ARBM-based method in the nonparallel setting. These results demonstrate that our method is especially effective in nonparallel settings.

Table 4.4: Average MCD [dB] on VCC 2018

	GMM	ARBM	NTD
Parallel	6.55		7.03
Nonparallel		7.97	7.70

4.4.6 Experiments on Speech with an Articulation Disorder

Finally, we conducted a VC experiment on speech with the articulation disorder resulting from the athetoid type of cerebral palsy. In this experiment, the source speaker is one male speaker with the articulation disorder and the target speaker is one physically unimpaired person. Fig. 4.9 depicts an example of the original spectrogram uttered by the person with articulation disorder and its converted spectrogram. As shown in this figure, the spectral power of the original speech is weak in high-frequency range, and that degrades the intelligibility of the speech. On the other hand, the middle- and high-frequency spectral power of converted speech is successfully emphasized by the proposed method.

4.5 Chapter Conclusions

An innovative dictionary-learning method of NMF-based voice conversion was proposed. It makes NMF-VC possible for nonparallel training. While exemplar-based VC retains the naturality of the converted speech to a high degree, the source and target dictionaries expand significantly. Although dictionary learning VC achieves compact dictionary representation, the parallel dictionaries of the source and target speakers are difficult to learn. These conventional NMF-VC methods require parallel utterances by the source and target speakers to construct the source and target dictionaries. In this study, a method parallel dictionary learning for NMF-VC based on NTD was proposed, that does not require parallel data during training. NTD decomposes an input observation into a set of mode matrices and one core tensor. In the proposed framework, it is assumed that NTD decomposes the spectrogram into the frequency basis matrix, phonemic

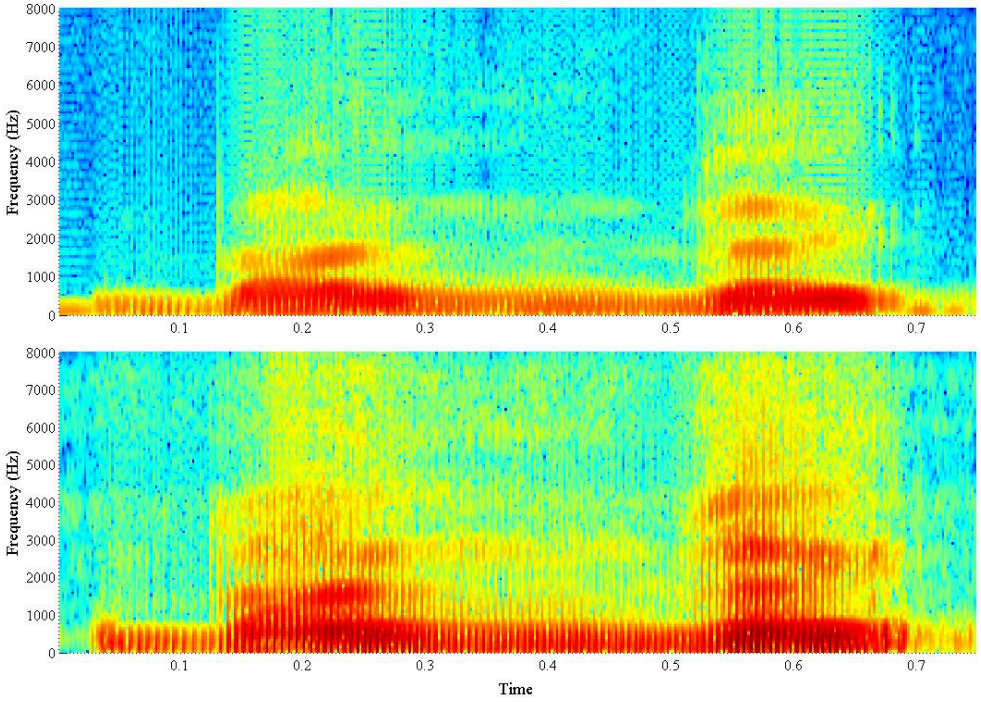


Figure 4.9: Example of the original spectrogram uttered by the person with articulation disorder (top) and its converted spectrogram (bottom). It is uttered/ b a n z a i/ ("hurrah" in English).

information matrix, and codebook matrix. Recently, several studies have been conducted for NMF-VC, and the scope of possible applications is widening. It is assumed that the proposed method assists these applications with nonparallel training. It was confirmed that the proposed method achieved an almost identical MCD to the conventional NMF-based dictionary learning that uses parallel data. Furthermore, the performance of the proposed method was comparable to that of the conventional ARBM-based method in a nonparallel setting.

In future work, we plan to apply the method to assistive technology for speakers with articulation disorders. The speech of such speakers is considerably different from that of the speech of unimpaired persons, and it is difficult to align correctly. The proposed method does not require the same texts of speech data for the source and target speakers or the framewise matching between acoustic features of both speakers. Furthermore, the NTD-based dictionary learning is a natural expansion of the NMF-based method, and it can read parallel and

nonparallel data to learn the dictionary. Therefore, we also aim to investigate a semi-supervised dictionary-learning method that improves the performance of a model trained with a small set of parallel data using a large set of nonparallel data.

In the real world, background noise deteriorates conversion performance. However, the proposed model has not been designed with noise robustness in mind. In order to retain the quality of converted voices in a noisy environment, noise robustness is required. In our previous study [94], a noise-robust NMF-based VC was proposed, where the performance was improved by 25% compared with the GMM-based method. As the currently proposed method is based on NMF-based VC, it will be easy to apply the noise-robust conversion. The evaluation of our proposed method for a noisy environment will be a topic for our future work.

Chapter 5

Knowledge Transferability between the Speech Data of Persons with Dysarthria Speaking Different Languages for Dysarthric Speech Recognition

The related publications for this chapter are [95].

5.1 The Motivation and Related Work

5.1.1 Motivation

In this chapter, we focused on the problem of speech recognition for persons with articulation disorders caused by the athetoid type of cerebral palsy. In the case of persons with this type of articulation disorders, it is difficult to collect sufficient speech data to train a model. However, most machine learning approaches require a large amount of training data. Moreover, a speaker-independent ASR system trained using the data of physically unimpaired persons is almost useless because their speech style differing significantly from that of physically unimpaired per-

5. KNOWLEDGE TRANSFERABILITY BETWEEN THE SPEECH DATA OF PERSONS WITH DYSARTHRIA SPEAKING DIFFERENT LANGUAGES FOR DYSARTHRIC SPEECH RECOGNITION

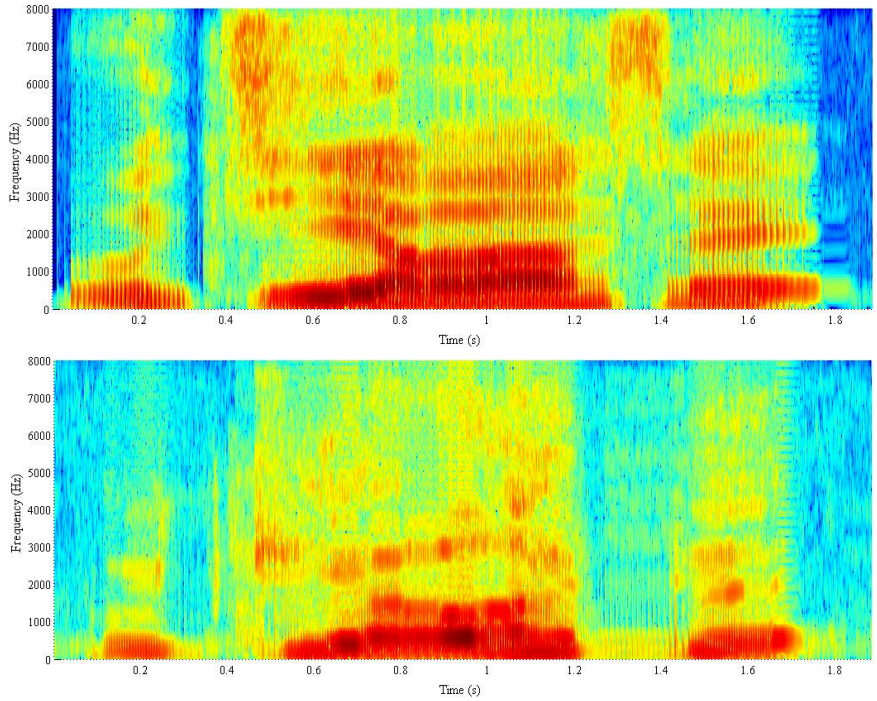


Figure 5.1: Example of spectrogram uttered for /u ch i a w a s e/ of a physically unimpaired person (top) and of a person with an articulation disorder (bottom). These spectrograms are stretched using dynamic time warping to be more easily observed.

sons. Therefore, to tackle these problems, we propose a transfer learning method using an additional speech database.

Their utterances are often unstable or unclear owing to athetoid symptoms. Fig. 5.1 depicts the spectrograms of a physically unimpaired person and of a person with an articulation disorder for the Japanese word "uchiawase" ("meeting" in English). As shown in this figure, the high-frequency spectral power of the person with an articulation disorder is weaker in comparison to that of the physically unimpaired person, demonstrating the unclear speech of persons with an articulation disorder, which makes their speech difficult to understand.

Automatic speech recognition (ASR) has been widely spread within services such as personal assistants on smartphones. In addition, remarkable progress has been made with respect to recent developments in deep learning for ASR [26, 96, 97] in fields with availability to a large amount of training data. However, there

has been no significant beneficial use of ASR achieved for persons with speech disorders as a result of differences in various speech styles and the limited amount of the training speech data available. In the case of persons with articulation disorders, owing to their speech style differing significantly from that of physically unimpaired persons, a speaker-independent ASR system trained using the data of physically unimpaired persons is almost useless. However, it is difficult to collect sufficient speech data from persons with articulation disorders to train the model. Therefore, there is a need for an approach specifically tailored to overcome the low data availability of impaired speech.

To solve the problem of limited data availability, we employ transfer learning [98], which seeks to apply the knowledge learned in one or more domains or tasks to another domain or task. To be specific, we use three different sources of data: speech data from a target speaker with an articulation disorder, speech data from physically unimpaired persons in the same target language, and speech data of persons with articulation disorders in other languages. In our previous work [99], a data-augmentation method based on an end-to-end ASR model was proposed. The model comprises a dysarthria-specific encoder, a physically unimpaired person-specific encoder, an English decoder, and a Japanese decoder. This method has exhibited the ability to efficiently integrate the knowledge gathered from the speech data of physically unimpaired Japanese-speaking persons, as well as Japanese-speaking and English-speaking persons with articulation disorders. However, it was difficult to train the model because it optimized all modules simultaneously.

In this paper, we investigate the knowledge transfer of the language-dependent characteristic corresponding to the speech of physically unimpaired persons as well as the language-independent characteristic of the dysarthric speech. We refer to a physically unimpaired Japanese-speaking person, an English-speaking person with an articulation disorder, and a Japanese-speaking person with an articulation disorder as JU, ED, and JD, respectively. As is widely known, a large amount of speech data of physically unimpaired persons is publicly available. We assume that knowledge of Japanese language-specific characteristics from JU speech data can be transferred and applied to JD speech. Moreover, several non-Japanese speech databases of persons with articulation disorders have been

published [100, 101, 102, 103, 104]. The speech impairment and abnormalities caused by dysarthria, such as imprecise consonants and distorted vowels, are common among speakers of different languages. Consequently, we assume that the characteristics of dysarthric speech are independent of language. According to these assumptions, we propose a method to transfer the knowledge learned from these rich speech data sources to the problem of Japanese dysarthric speech recognition, for which training data is insufficient. Utilization of the speech data from physically unimpaired persons who speak the same language as the target speaker has been considered in some literatures [6, 105, 106]. However, these studies only considered the linguistic characteristics within the language. Following the assumption that the characteristics of dysarthric speech are language-independent, our work utilizes data from persons with articulation disorders who speak different languages in addition to using data from persons who speak the same language as the target speaker. Compared to the random initialization without using additional data, our proposed approach provides a considerable reduction in phoneme error rate.

5.1.2 Related Work

Previously, we have published several research works on speech recognition for JDs using speech data which was collected using our own methods. Instead of using discrete cosine transform, we proposed robust feature extraction based on principal component analysis [107], which provides more stable utterance data. In [108], multiple acoustic frames as an acoustic dynamic feature to improve the speech recognition rate of a person with dysarthria, particularly for speech recognition using dynamic features only. We proposed feature extraction based on a convolutional neural network to process small local fluctuations of speech uttered by a person with an articulation disorder [109]. These methods handle feature extraction and do not account for the limited amount of speech data specific to persons with articulation disorders.

Voice conversion (VC) is an approach that can be used to mitigate the problem of limited data availability. Aihara *et al.* [110] have proposed a VC method

Table 5.1: Popular dysarthric speech databases.

Database	Description	# Speaker	# Utterance	Type
Nemours [100]	Short sentence	11	814	CP
UA speech corpus [101]	Word	19	14,535	CP
TORGO [102]	Word, Sentence	8	3,245+	CP, ALS

CP: cerebral palsy
ALS: amyotrophic lateral sclerosis

based on partial least square using a phoneme-discriminative feature that converts a dysarthric voice into non-dysarthric speech. Jiao *et al.* [6] have proposed a data-augmentation method based on convolutional generative adversarial network (GAN) [111]. In [112], speech synthesis-based data augmentation was proposed. The dysarthric-like speech was generated using temporal and speed modifications applied to speech of physically unimpaired persons, and this generated speech dataset was used thereafter to train an ASR based on a deep neural network. Vachhani *et al.*[105] have proposed a feature enhancement method based on an autoencoder. This method implies training the autoencoder with the use of a speech signal obtained from a physically unimpaired control speaker, after which this speech data was used to convert dysarthric speech into an improved feature representation.

Several public databases are available for clinical speech applications [100, 101, 102] as shown in Table 5.1. Several researchers have worked on developing an ASR system using these databases [113, 114, 115]. However, speakers included in these databases are English speakers, and there is no publicly available database with speech data obtained from Japanese speakers. Therefore, establishing an ASR for Japanese speakers with articulation disorders is very challenging.

Knowledge transferability across different languages allows performance improvements for a multilingual ASR system and is especially efficient for low-resource languages. Considering multilingual speech recognition tasks, Toshniwal *et al.* [116] have jointly trained a single ASR model across a dataset composed of data corresponding to nine Indian languages; this approach has shown improve-

ments over monolingual models. This suggests that an ASR model can provide richer internal representation across several languages. To train a model, we proposed an end-to-end speech recognition framework [99] that uses the speech data of JUs, EDs, and a JD. This approach suggested a means to transfer knowledge across a language for speech affected by dysarthria; however, the difficulty related to model training remained. In contrast, the new method proposed in this paper allows training a model easily on both the dysarthria-specific and Japanese-specific acoustic characteristics.

5.2 Listen, attend and spell model

In this work, we employ an end-to-end ASR model for dysarthric speech recognition. Previous works on ASR have proposed various end-to-end learning models combining acoustic and language models within a sequence-to-sequence framework [61, 117]. Unlike ASR systems based on traditional hidden Markov models, these models learn all components of the ASR system jointly. Therefore, it enables the development of ASR systems for new applications and configurations. In this work, we investigate an end-to-end ASR model based on the LAS model [118] for dysarthric speech.

The LAS model [118] consists of a listener module and a speller module which are trained jointly. The goal of this model is to generate the probability of a grapheme sequence based on information from the previous graphemes and a sequence of acoustic features. The model can be defined as follows:

$$P(\boldsymbol{y}|\boldsymbol{x}) = \prod_{s} P(y_s|\boldsymbol{x}, \boldsymbol{y}_{<s}), \tag{5.1}$$

where $\boldsymbol{x} = (x_1, ..., x_t, ..., x_T)$ and $\boldsymbol{y} = (y_1, ..., y_s, ..., y_S)$ denote sequences of acoustic features and graphemes, respectively. Here, x_t, y_s, T, and S denote the input acoustic feature frame, the posterior distribution of the output grapheme, the number of the input acoustic features, and the output graphemes respectively. A listener is an encoder-recurrent neural network (RNN) that transforms an input sequence $\boldsymbol{x}$ of acoustic features into a high level representation $\boldsymbol{h} = (h_1, ..., h_u, ..., h_U)$, where h_u and $U \leq T$ are the encoder output feature and

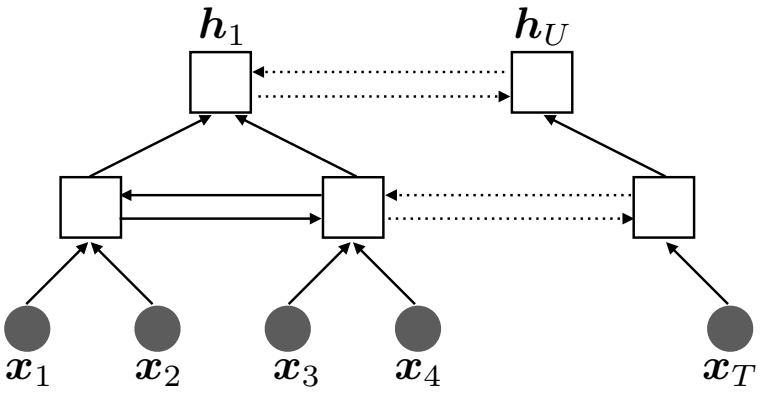


Figure 5.2: Network structure of the listener. Each layer has a pyramid structure that takes every two consecutive frames of the output from the previous layer as input.

the number of the encoder output sequence, respectively. A speller is a decoder RNN that consumes $\boldsymbol{h}$ and produces a probability distribution over grapheme sequence $\boldsymbol{y}$.

The listener is organized as a stacked pyramid bidirectional long short-term memory (pBLSTM) as shown in Fig. 5.2. The pyramid structure allows for reducing the computational complexity and the convergence time and provides the speller with the ability to extract the relevant information within a smaller number of time steps. The listener is considered as the acoustic model in an ASR system. Its operation is defined as follows:

$$\boldsymbol{h} = \text{Listen}(\boldsymbol{x}; \theta_{Lis}), \tag{5.2}$$

where θ_{Lis} denotes the parameters of the listener.

The speller is an attention-based long short-term memory (LSTM) transducer, which is organized as a stacked unidirectional RNN. At each time step, the speller produces a probability distribution over the subsequent graphemes conditioned on all the graphemes obtained previously. The attention mechanism allows the speller to generate the next output over graphemes encapsulating information within the acoustic signal. The speller is considered as a language model in an

ASR system. The speller operation is written as follows:

$$P(\boldsymbol{y}|\boldsymbol{x}) = P(\boldsymbol{y}|\boldsymbol{h};\theta_{Spl}) = \prod_{s} P(y_s|\boldsymbol{h},\boldsymbol{y}_{<s};\theta_{Spl}) \tag{5.3}$$

$$= \mathrm{Spell}(\boldsymbol{h};\theta_{Spl}), \tag{5.4}$$

where θ_{Spl} denotes the parameters of the speller.

The model is trained to optimize the discriminative loss as follows:

$$\mathcal{L}(\mathcal{D},\theta_{Lis},\theta_{Spl}) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\mathcal{D}}[-\log(P(\boldsymbol{y}|\boldsymbol{x}))]. \tag{5.5}$$

Here, $\mathcal{D}$ denotes the joint distribution over input sequence $\boldsymbol{x}$ and label sequence $\boldsymbol{y}$.

5.3 Proposed Method

In this section, we explain two knowledge transfer methods, using speech data obtained from JUs and EDs, which could be used for speech recognition for a JD. Our proposed ASR system is based on the LAS model. Considering the JD dataset, let $\mathcal{D}_{JD}$ be the joint distribution over the input sequence and the corresponding label sequence. $\mathcal{D}_{ED}$ and $\mathcal{D}_{JU}$ are analogously defined for the EDs and JUs datasets, respectively. In this work, the speller produces a phoneme of the corresponding language.

5.3.1 Transfer learning using the speech data obtained from the physically unimpaired persons

In this section, we explain the intra-language knowledge transfer from the speech data obtained from physically unimpaired persons to a speech representation for a person with an articulation disorder. Fig. 5.3 depicts the overview of this pre-training scheme. First, we pre-train a single LAS model using the speech data obtained from physically unimpaired persons while adjusting the parameters to minimize the loss function as follows:

$$\hat{\theta}_{Lis},\hat{\theta}_{Spl} = \underset{\theta_{Lis},\theta_{Spl}}{\arg\min}\,\mathcal{L}(\mathcal{D}_{JU},\theta_{Lis},\theta_{Spl}), \tag{5.6}$$

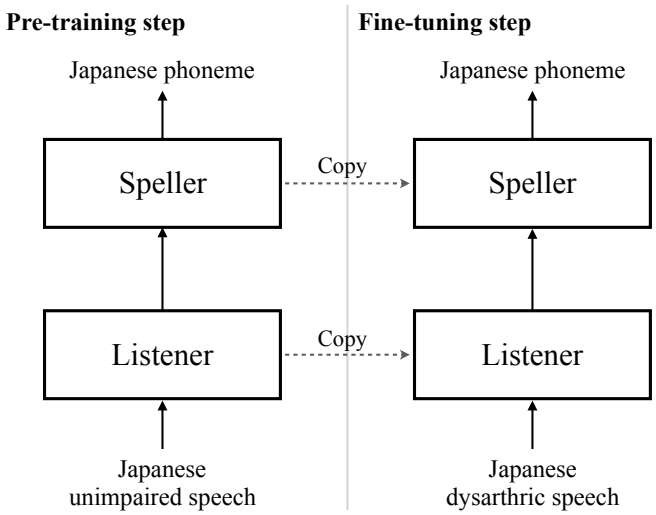


Figure 5.3: Training scheme of a single LAS model with pre-training using the speech of JUs.

where $\hat{\theta}_{Lis}$ and $\hat{\theta}_{Spl}$ are the optimized parameters of the listener and speller, respectively. It is assumed that these pre-trained parameters have appropriate representation for producing Japanese phonemes. Then, obtained parameters $\hat{\theta}_{Lis}$ and $\hat{\theta}_{Spl}$ are fine-tuned using the speech data taken from a person with an articulation disorder, optimized as follows:

$$\underset{\hat{\theta}_{Lis},\hat{\theta}_{Spl}}{\arg\min}\, \mathcal{L}(\mathcal{D}_{JD}, \hat{\theta}_{Lis}, \hat{\theta}_{Spl}). \tag{5.7}$$

At the prediction step, we use the well-trained listener and speller parameters to produce Japanese phonemes.

Inspired by our previous work [99], we constructed another model that consists of two listeners and one speller for fine-tuning as shown in Fig. 5.4. One is a dysarthria-specific listener called "D-Listener", and the other is a physically unimpaired speaker-specific listener called "U-Listener". This model is also trained for optimizing parameters as follows:

$$\mathcal{L}(\mathcal{D}_{JD}, \theta_{D\text{-}Lis}, \hat{\theta}_{Spl}) + \mathcal{L}(\mathcal{D}_{JU}, \theta_{U\text{-}Lis}, \hat{\theta}_{Spl}), \tag{5.8}$$

where $\theta_{D\text{-}Lis}$ and $\theta_{U\text{-}Lis}$ denote parameters of D-Listener and U-Listener and are initialized to $\hat{\theta}_{Lis}$ before training. In this fine-tuning, the speller is initialized

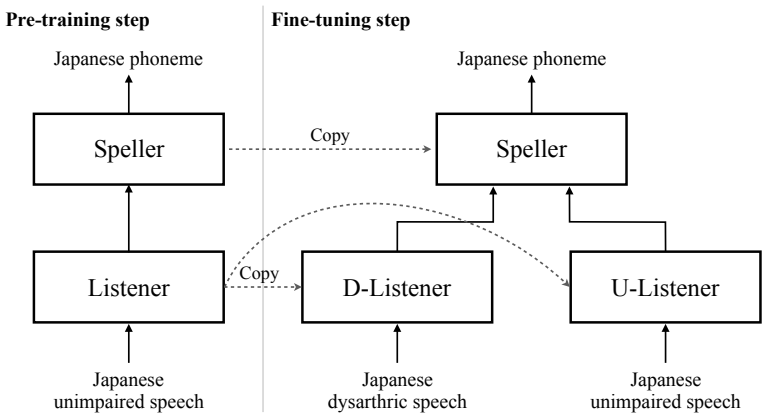


Figure 5.4: Training scheme of a two-encoder LAS model with pre-training using the speech of JUs. The speech data of physically unimpaired persons are used to explicitly separate dysarthria characteristics during the fine-tuning step.

to $\hat{\theta}_{Spl}$ and is shared between a person with an articulation disorder and the physically unimpaired persons. The acoustic characteristic of dysarthric speech considerably differs from that of the speech of a physically unimpaired person owing to the athetoid symptoms. Therefore, we use the dysarthria-specific listener for the speech data of JD.

5.3.2 Transfer learning using multilingual speech data obtained from a person with dysarthria

In this section, we explain the knowledge transfer from the speech data obtained from both JUs and EDs to a speech representation for JD. First, we configure a shared listener called "S-Listener", a Japanese speller called "J-Speller", and an English speller called "E-Speller", as shown in Fig. 5.5. The S-Listener is shared between EDs and JUs. This mechanism is similar to multitask learning [119]. Multitask learning simultaneously executes multiple tasks for one input data. However, our proposed approach differs in that each input data from different domains is processed to estimate the phonemes of only the corresponding language. E-Speller and J-Speller produce phoneme sequences in English and Japanese, respectively. This model is optimized by adjusting the parameters to

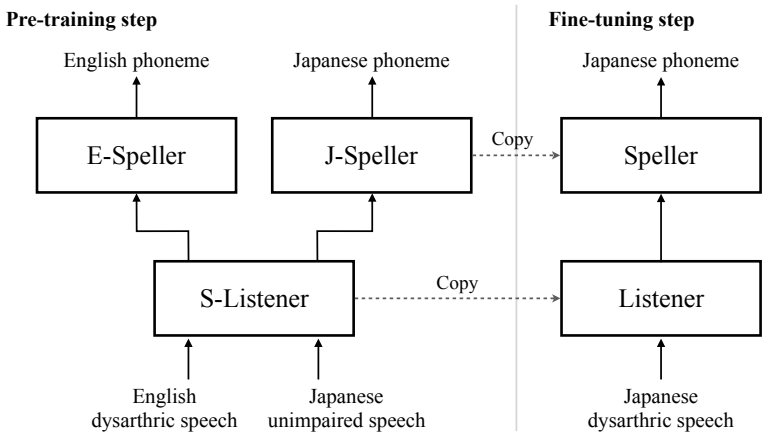


Figure 5.5: Training scheme of a single LAS model with pre-training using the speech of JUs and the speech of EDs. In the pre-training step, the speller is switched according to the input language.

minimize the loss function as follows:

$$\mathcal{L}(\mathcal{D}_{ED}, \theta_{S\text{-}Lis}, \theta_{E\text{-}Spl}) + \mathcal{L}(\mathcal{D}_{JU}, \theta_{S\text{-}Lis}, \theta_{J\text{-}Spl}), \tag{5.9}$$

where $\theta_{S\text{-}Lis}$, $\theta_{E\text{-}Spl}$, and $\theta_{J\text{-}Spl}$ denote parameters of S-Listener, E-Speller, and J-Speller, respectively. All components are learned jointly. We expect that this cross-lingual mechanism will help the listener module capture a better high-level representation containing both the dysarthria-specific characteristic and the expression capability of Japanese. Then, the obtained parameters $\theta_{S\text{-}Lis}$ and $\theta_{J\text{-}Spl}$ are jointly fine-tuned using speech data of JD with an articulation disorder, as defined in (5.7).

5.4 Experiments

In this work, we conducted experiments on the speaker-dependent system for each target speaker with dysarthria.

5.4.1 Experimental Conditions

Our proposed approach was evaluated on a phoneme recognition task suggested to five Japanese-speaking males with articulation disorders. We repeatedly recorded

Table 5.2: Dataset statistics gathered for JDs.

Speaker	# words	# phonemes	# repetitions	# utterances
JM1	204	1,639	3	612
JM2	210	1,687	5	1,050
JM3	216	1,731	5	1,080
JM4	215	1,721	3	645
JM5	213	1,705	3	639

216 words included in the ATR Japanese speech database [90] for each speaker as shown in Table 5.2. The number of repetitions differed for each speaker owing to the athetoid symptoms. In our experiments, the first utterances of each word were used for evaluation, and the other utterances (e.g., 864 words for JM3) were used to train models. In the experiment, JUs were five male and five female speakers, whose speech is stored in the ATR Japanese speech database. We used the same 216 words (1,731 phonemes) for each speaker as in the dysarthric dataset. Considering the speech dataset corresponding to EDs, we used the TORGO database [102], which includes three female and five male persons. This database has missing data owing to a clipping error; therefore, we selected usable word speech as shown in Table 5.3. When we pre-trained (in all proposed methods) or fine-tuned (only in the two-encoder LAS) modules using JU speech and ED speech, we used all speakers' speech. When we fine-tuned modules using Japanese dysarthric speech for a speaker-dependent model, we used only the target speaker's speech. In this manner, for each Japanese dysarthric subject, we trained the speaker-dependent model and evaluated the model independently. We used 39-dimensional mel-frequency cepstral coefficient (MFCC) features (13-order MFCCs, their delta, and acceleration) as the input features, computed every 10 ms over a 25 ms window.

Considering the listener configuration, we used 2 layers of 512 pBLSTM nodes (256 nodes per direction). For the speller configuration, we used a one-layer LSTM with 512 nodes. In this work, we used the phoneme sequence as the output sequence. The numbers of phonemes for English and Japanese were 57 and 54, respectively. The network was optimized using an Adam optimizer [120]

Table 5.3: Dataset statistics gathered from the TORGO database for EDs.

Speaker	# phonemes	# utterances
F01	1,014	188
F03	5,290	777
F04	3,531	489
M01	4,217	556
M02	4,086	568
M03	3,860	594
M04	3,519	499
M05	3,346	443

Table 5.4: Phoneme error rates [%] estimated for each method. Jpn and Eng denote Japanese and English respectively. All systems are based on the target speaker-dependent models.

	Pre-training		Fine-tuning		Speaker					
	Architecture	Database	Architecture	Database	JM1	JM2	JM3	JM4	JM5	mean
rand init	-	-	single LAS	JD	48.70	19.29	21.56	53.75	49.16	38.49
multi	-	-	single LAS	JD & JU	40.37	22.53	18.81	49.67	49.72	36.22
trans. 1	single LAS	JU	single LAS	JD	35.34	18.40	11.06	41.63	45.24	30.33
trans. 2	single LAS	JU	two-encoder LAS	JD & JU	35.09	17.17	10.42	40.32	43.74	29.35
trans. 3	two-decoder LAS	JU & ED	single LAS	JD	26.37	15.95	9.66	33.86	42.60	25.69

with label smoothing [121]. We constructed one batch with sub-batches of 64 samples for each domain. The number of epochs was 500, and the learning rate was set to 1e-4.

For the baseline system, we trained two models based on the conventional single LAS model. The first model was trained using only speech data of JUs ('rand init'), and the second model was trained using the data of both JUs and JD ('multi').

5.4.2 Experimental Results

Table 5.4 lists the phoneme error rates (PERs) corresponding to each method. In this table, a lower PER indicates a better result. Here, 'trans. 1' and 'trans. 2'

are the transfer learning methods using the speech data of only JUs, and 'trans. 3' is the transfer learning method using the speech data of both JUs and EDs.

Multi-condition learning using the speech data of JUs ('multi') achieved a slightly lower PER score than the speaker-dependent model with the random initialization ('rand init'). However, for speakers JM2 and JM5, the performance deteriorated. This result indicates the need for proper initialization and integration of the different data domains.

It is possible to observe the effects of knowledge transfer from the speech data of other domains. Pre-training using the speech data of JUs ('trans. 1') achieved 21.2% and 16.3% average relative improvements compared with 'rand init' and 'multi', respectively. Moreover, fine-tuning with two encoders ('trans. 2') slightly outperformed 'trans. 1'.

Compared with the random initialization, our proposed transfer learning method using speech obtained from JUs and English dysarthric speech ('trans. 3') achieved a 33.3% relative improvement. Furthermore, we obtained a significant improvement of 15.3% relative PER compared to pre-training excluding speech data obtained from EDs.

5.4.3 Discussion

We assume that the ability to recognize Japanese dysarthric speech can be transferred from the combination of dysarthric speeches in other languages with physically unimpaired Japanese speech. The proposed approach achieved significantly better performance than the random initialization. These results indicate that the language transferability as provided in [116] is effective even in the case of dysarthric speech. Additionally, we obtained significant improvements even if the speaker had small amounts of speech data. For example, in the case of speaker JM1, our proposed approach achieved a 45.9% relative improvement compared with the random initialization. As the speech data of speakers with articulation disorders is quite limited, this effect is deemed to be crucial for the purposes of the present research.

5.5 Chapter Conclusions

In this chapter, we proposed a novel knowledge transfer approach for dysarthric speech recognition that uses speech data obtained both from physically unimpaired persons and from persons with dysarthria speaking in a different language. The amount of speech data obtained from speakers with articulation disorders is quite limited owing to the athetoid symptoms. To solve this problem, we used additional speech data obtained from physically unimpaired persons speaking in a different language. We demonstrated the effectiveness of the proposed approaches through the phoneme recognition task.

Our future research will further investigate the usage of speech data published in the publicly available databases obtained from persons with dysarthria speaking in other languages. This research will then also investigate increasing the amount of the speech data of JUs and persons with dysarthria who do not speak Japanese. Moreover, we will apply our proposed approach to the conventional deep neural network-hidden Markov model hybrid ASR system to compare the results of the proposed methods against the previously proposed frameworks [113]. It should be noted that, in this work, we use the phoneme sequence to train the model. However, dysarthric speech may contain undesirable errors with respect to an expected phoneme sequence owing to the athetoid symptoms. In such cases, the given training phoneme sequence would be unreliable. To solve this problem, we will investigate approaches to apply unsupervised domain adaptation. Domain adaptation using deep learning has been researched and has shown the remarkable progress. Therefore, we expect the application of a domain adaptation technique to dysarthric ASR to improve performance.

Chapter 6

Audio-Visual Speech Recognition Using Convolutive Bottleneck Networks for a Person with Severe Hearing Loss

The related publications for this chapter are [122].

6.1 The Motivation and Related Work

6.1.1 Motivation

In this chapter, we focused on the problem of speech recognition for persons with articulation disorders resulting from severe hearing loss. Similar to articulation disorder resulting from athetoid cerebral palsy, the speech style of a person with this type of articulation disorder is also different from those of people without hearing loss. Thus, a speaker-independent ASR model for unimpaired persons is hardly useful in recognizing such speech, especially in noisy environments. To solve this problem, we propose audio-visual (multimodal) ASR system using the lip movement.

In recent years, a number of assistive technologies using information processing have been proposed; for example, sign language recognition using image

recognition technology [123, 124] and text reading systems from natural scene images [125]. Among them, in this chapter, we focus on an assistive technology for a person with hearing loss. In Japan alone, there are 360,000 people.

Some people with hearing loss who have received speech training or who lost their hearing after learning to speak can communicate using spoken language. However, in the case of automatic speech recognition (ASR), their speech style is so different from that of people without hearing loss that a speaker-independent (audio-visual) ASR model for unimpaired persons is hardly useful in recognizing such speech. Matsumasa *et al.* [126] researched an ASR system for articulation disorders resulting from cerebral palsy and revealed the same problem.

For people with hearing problems, lip reading is one communication skill that can help them communicate better. In the field of speech processing, audio-visual speech recognition has been studied for robust speech recognition under noisy environments [127, 128, 129]. In this paper, we propose an audio-visual speech recognition for articulation disorders resulting from severe hearing loss.

The main contribution of this paper is that we propose a bottleneck feature extracted from audio-visual features. Convolutive Bottleneck Network (CBN) [130], which stacks multiple layers of various types (such as a convolution layer, a subsampling layer, and a bottleneck layer) [131, 132] forming a deep network, is applied to audio-visual data. The bottleneck layer reduces the number of units for the adjacent layers, and, consequently, we can expect that each unit in the bottleneck layer aggregates information and behaves as a compact feature descriptor that represents an input with a small number of bases.

Our experimental results confirmed that our bottleneck features have robustness for small local fluctuations that are caused by the utterances of those who have hearing loss. Moreover, our integration of audio and visual features acquired robustness in noisy environments.

6.1.2 Related Work

As one of the techniques used for robust speech recognition under noisy environments, audio-visual speech recognition, which uses lip dynamic visual information and audio information, has been studied. In audio-visual speech recognition, there

are mainly three integration methods: early integration [127], which connects the audio feature vector with the visual feature vector; late integration [129], which weights the likelihood of the result obtained by a separate process for audio and visual signals; and synthetic integration [128], which calculates the product of output probability in each state.

In audio-visual speech recognition, detecting face parts (for example, eyes, mouth, nose, eyebrows, and outline of face) is an important task. The detection of these points is referred to as face alignment. The Active Appearance Model (AAM) [133] and Active Shape Model (ASM) [134] are well-known face alignment models. In this paper, we employed a Constrained Local Model (CLM) [135, 136]. A CLM is a subject-independent model that is trained from a large number of face images.

In recent years, an ASR system has been applied as assistive technology for people with articulation disorders. During the last decades, we have researched an ASR system for a person with cerebral palsy. In [126], we proposed robust feature extraction based on principal component analysis (PCA) with more stable utterance data instead of discrete cosine transform (DCT). In [108], we used multiple acoustic frames (MAF) as an acoustic dynamic feature to improve the recognition rate of a person with an articulation disorder, especially in speech recognition using dynamic features only.

Deep learning has had recent successes for acoustic modeling [137]. Deep Neural Networks (DNNs) contain many layers of nonlinear hidden units. The key idea is to use greedy layer-wise training with Restricted Boltzmann Machines (RBMs) followed by fine-tuning. Ngiam *et al.* [138] proposed multimodal DNNs that learn features over audio and visual modalities.

In this paper, we employ a Convolutional Neural Network (CNN) [131, 132]-based approach to extract robust features from audio and visual features. The CNN is regarded as a successful tool and has been widely used in recent years for various tasks, such as image analysis [139] and spoken language [140]. In [46], CNN is employed as robust feature extraction for the fluctuation of the speech uttered by a person with cerebral palsy. Experimental results in [46] revealed that the convolution and pooling operations in CNN have a robustness to the

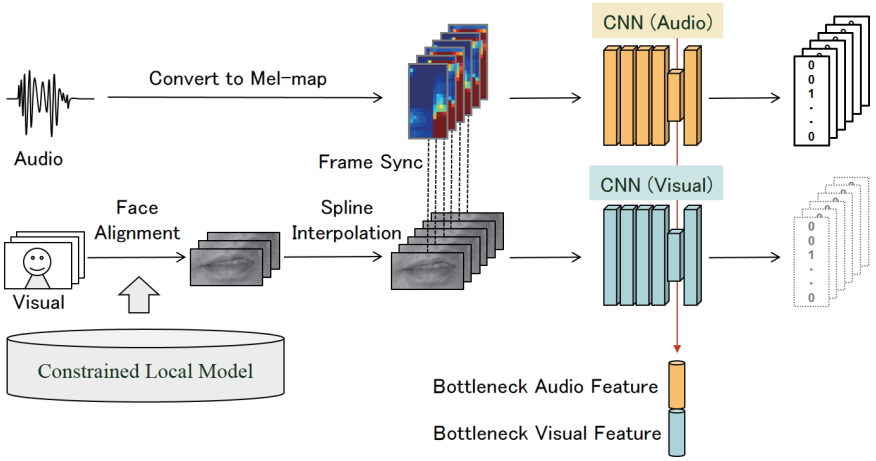


Figure 6.1: Flow of the feature extraction.

small local fluctuation which is caused by motor paralysis resulting from athetoid cerebral palsy.

6.2 Proposed Method

Fig. 6.1 shows the flow of our proposed feature extraction. First, we prepare the input features for training a CBN from audio and visual signals. For the audio signals, after calculating short-term mel spectra from the signal, we obtain mel-maps by dividing the mel spectra into segments with several frames, allowing overlaps.

The visual signals of the eyes, mouth, nose, eyebrows, and outline of the face are aligned using a Constrained Local Model (CLM) and a lip image is extracted. The details of lip image extraction are explained in the following section. The extracted lip image is interpolated to fill the sampling rate gap between audio features.

For the output units of the CBN, we use phoneme labels that correspond to the input mel-map and lip images. Audio and visual CBN are separately trained. The input mel-map and lip images are converted to the bottleneck feature by using each CBN. Extracted features are used as the input feature of Hidden Markov Models (HMM).

6.2.1 Lip Image Extraction Using CLM

Face alignment of this paper is conducted by using the Point Distribution Model (PDM) and its model parameter is estimated by CLM. CLM consists of two steps. The first step is the face point detection and the second step is parameter estimation.

6.2.2 PDM

We model a facial image of a large number of people by using the PDM which models a facial image by 2-dimensional shape vectors. The position vector which corresponds to the point of the PDM is defined as follows:

$$\mathbf{X} = (\mathbf{X}_1^{\mathrm{T}}, ..., \mathbf{X}_M^{\mathrm{T}})^{\mathrm{T}} \tag{6.1}$$

where $\mathbf{X}_i = (x_i, y_i)^{\mathrm{T}}$ and M denote the i-th point of PDM and the number of points of PDM, respectively. The position vector is represented as follows:

$$\mathbf{X} = \overline{\mathbf{X}} + \mathbf{\Phi q} \tag{6.2}$$

where $\mathbf{\Phi}$, $\mathbf{q}$ and $\overline{\mathbf{X}}$ denote the principal vectors extracted by Principal Component Analysis (PCA), the parameter vector and the mean vector of the shape vector, respectively. By using PDM, the i-th point on the image, $\mathbf{X}_i(\mathbf{p})$, is represented as follows:

$$\mathbf{X}_i(\mathbf{p}) = s\mathbf{R}[\overline{\mathbf{X}}_i + \mathbf{\Phi_i q}] + \mathbf{t} \tag{6.3}$$

where $\mathbf{p} = \{s, \mathbf{R}, \mathbf{t}, \mathbf{q}\}$ denotes the parameter set. s denotes a scale and $\mathbf{R}$ denotes a rotation which consists of pitch α, yaw β, roll γ. $\mathbf{t}$, $\mathbf{q}$ and $\mathbf{\Phi_i}$ denote the shift vector, the parameter vector and the i-th principal vector, respectively.

6.2.3 CLM

The parameter of PDM is estimated by using CLM. First, feature points are detected by Support Vector Machine (SVM) which is trained by a large number of facial images.

Then, the model parameter $\mathbf{p}$ is estimated from the i-th detected feature point $\hat{\mathbf{X}}_i$ by minimizing the following equation:

$$Q(\mathbf{p}) = \sum_{i=1}^{M} \|\hat{\mathbf{X}}_i - \mathbf{X}_i(\mathbf{p})\|^2 + R(\mathbf{p}) \tag{6.4}$$

where $R(\mathbf{p})$ is a regularization term to avoid over fitting. In this paper, we defined $R(\mathbf{p})$ as normal distribution of $N(0, \mathbf{\Lambda})$.

6.2.4 Feature Extraction Using CBN

6.2.4.1 Convolutive bottleneck network

A CBN consists of an input layer, a pair of a convolution layer and a pooling layer, fully-connected Multi-Layer Perceptrons (MLPs) with a bottleneck structure, and an output layer as shown in Fig. 6.2. C, S, and M denote convolutional layer, sub-sumpling layer, and MLPs, respectively. The MLP shown in Fig. 6.2 stacks three layers (M1 , M2 , M3), and the number of units in the middle layer (M2) is reduced as "bottleneck features". The number of units in each layer is discussed in the experimental section. Since the bottleneck layer has reduced the number of units for the adjacent layers, we can expect that each unit in the bottleneck layer aggregates information and behaves as a compact feature descriptor that represents an input with a small number of bases, similar to other feature descriptors, such as MFCC, Linear Discriminant Analysis (LDA) or PCA. In this paper, audio and visual features are input to each CBN and extracted bottleneck features are used for multimodal speech recognition.

6.2.4.2 Bottleneck feature extraction

First, we train audio and visual CBN. We prepare the input features for training a CBN from an image and speech signal uttered by person with hearing loss. For the audio feature, we obtain mel-maps by dividing the mel spectra into segments with several frames, allowing overlaps. For the output units of the CBN, we use phoneme labels that correspond to the input mel-map. For example, when we have a mel-map with the label /i/, only the unit corresponding to the label /i/

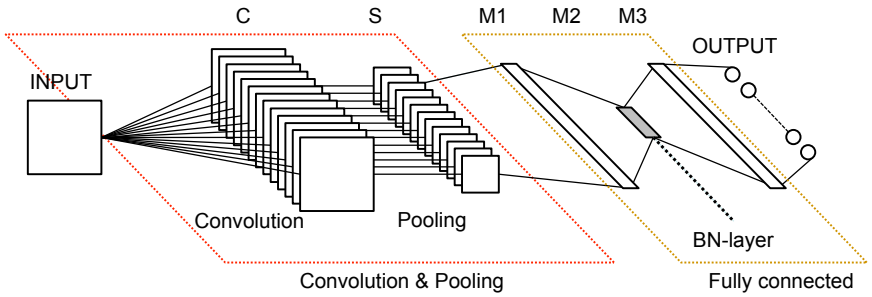


Figure 6.2: Convolutional Bottleneck Network.

is set to 1, and the others are set to 0 in the output layer. The label data is obtained by forced alignment using HMMs from the speech data.

For the visual features, because its sampling rate is smaller than the audio signal, spline interpolation is adopted to the images in order to fill the sampling rate gap. The output units of the CBN are the same as that of the audio features.

The parameters of the CBN are trained by back-propagation with stochastic gradient descent, starting from random values. The bottleneck (BN) features in the trained CBN are then used in the training of an HMM for speech recognition. In the test stage, we extract features using the CBN, which tries to produce the appropriate phoneme labels in the output layer. Again, note that we do not use the output (estimated) labels for the following procedure, but we use the BN features in the middle layer, where it is considered that information in the input data is aggregated. Finally, extracted bottleneck audio and visual features are used as the input features of audio or visual HMMs and the recognition results are integrated. Details about this integration are discussed in section 6.3.3.

6.3 Experiments

6.3.1 Experimental Conditions

Our proposed method was evaluated on word recognition tasks for one male person with hearing loss. We recorded 216 words included in the ATR Japanese speech database B-set which are used as test data and 2,620 words included in the ATR Japanese speech database A-set which are used as training data. The

utterance signal was sampled at 16 kHz and windowed with a 25-msec Hamming window every 10 msec. For the acoustic model, we used the monophone-HMMs (54 phonemes) with 5 states and 6 mixtures of Gaussians. For the visual model, we used the monophone-HMMs (54 phonemes) with the same states and mixtures of Gaussians to the acoustic model. The number of units of bottleneck features is 30. Therefore, input features of HMM are 30-dimensional acoustic features and 30-dimensional visual features. We compare our bottleneck feature with conventional MFCC+ΔMFCC (30-dimensions). Furthermore, we evaluated our method in noisy environments. We added white noise to audio signals and its SNR is set to 20dB, 10dB, and 5dB. Audio CBN and HMMs are trained by using the clean audio feature.

6.3.2 Architecture of CBN

As shown in Fig. 6.2, we use deep networks which consist of a convolution layer, a pooling layer and fully-connected MLPs. For the input layer of audio CBN, we use a mel-map of 39-dimensional-melspectrum $\times$ 13, and the frame shift is 1. For the input layer of visual CBN, frontal face videos are recorded at 60 fps. Luminance images are extracted from the image by using CLM and resized to 12×24 pixels. Finally, the images are up-sampled by spline interpolation and input to the CBN.

Table 6.1 shows the size of each feature map. The numbers of units in each layer of MLPs are set to 108, 30, 54. Those numbers are the same to audio CBN and visual CBN.

Table 6.1: Size of each feature map. $(k, i \times j)$ indicates that the layer has k maps of size $i \times j$.

	Input	C1	S1
Audio CBN	1, 39×13	13, 36×12	13, 12×4
Visual CBN	1, 12×24	13, 8×20	13, 4×10

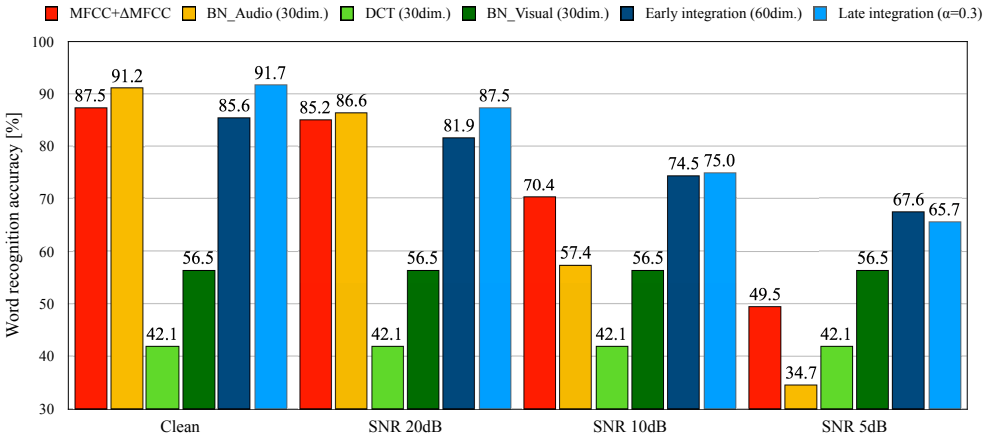


Figure 6.3: Word recognition accuracy using HMMs.

6.3.3 Experimental Results

Compared input features for the HMMs are listed as follows:

- MFCC+ΔMFCC
- Audio Bottleneck features (BN Audio)
- Discrete Cosine Transformation (DCT)
- Visual Bottleneck features (BN Visual)
- Early integration of BN Audio and BN Visual
- Late integration of BN Audio and BN Visual

In early integration, an audio feature and a visual feature are combined into a single frame and this frame used as an input feature for the HMMs. In late integration, an audio feature and a visual feature are input to each audio and visual HMM, and the output likelihood is integrated as follows:

$$L_{A+V} = \alpha L_V + (1-\alpha) L_A , \quad 0 \leq \alpha \leq 1 \tag{6.5}$$

where L_{A+V}, L_A, L_V and α denote integrated likelihood, likelihood of an audio feature, likelihood of a visual feature, and weights of likelihood, respectively.

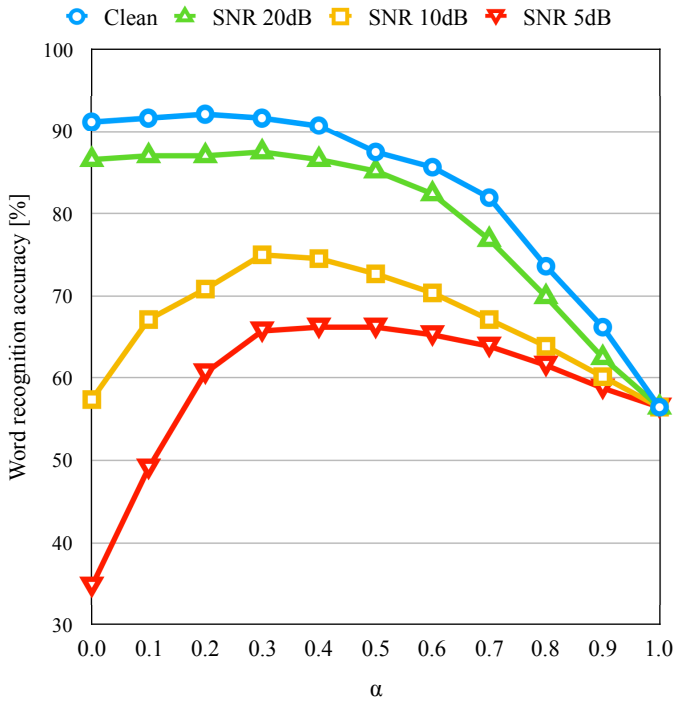


Figure 6.4: Word recognition accuracy using HMMs on the late integration.

Fig. 6.3 shows the word recognition accuracies in noisy environments. The bottleneck audio feature shows the best results compared to conventional MFCC at the clean environment and SNR of 20dB. This is due to the robustness of the CBN features to small local fluctuations in a time-mel-frequency map, caused by the articulation disordered speech.

The word recognition rate of lip reading using the bottleneck visual feature is 50.9%. At the SNR of 10dB, the early integration between audio and visual bottleneck features improved 4.1% from our baseline. Moreover at the SNR of 5dB, the early integration between audio and visual bottleneck features improved 18.1% from our baseline. t can be seen from these results that multimodal features are shown to be effective in noisy environments.

Fig. 6.4 shows the word recognition accuracies in the evaluation set as a function of the weight of the likelihood (α in (6.5)). $\alpha = 0.0$ in Fig. 6.4 shows the result of ASR using audio features only and $\alpha = 1.0$ in Fig. 6.4 shows the result of lip reading. This figure shows the best value for α under each condition. At

the SNR of 10dB and SNR 5dB, the graph is convex, and these results show the effectiveness of multimodal features in noisy environments.

6.4 Chapter Conclusions

We proposed multimodal bottleneck features using CBN for articulation disorders resulting from severe hearing loss. Compared with conventional MFCC, our proposed audio bottleneck feature shows the better results. We assume that is because our bottleneck features are robust to small local fluctuations, which are caused by hearing loss. In noisy environments, our proposed method using multimodal bottleneck features shows its effectiveness in comparison to the other methods. Since the tendency of the fluctuations in articulation disordered speech depend on the speaker, we would like to apply and investigate our method to a variety of speakers with speech disorders in the future.

Chapter 7

Conclusions

This thesis focuses on assistive technologies for persons with two types of articulation disorders: cerebral palsy and severe hearing loss. The common problem of these symptoms is their speech styles are quite different from those of unimpaired persons. Therefore, the speech processing application for typical voice is hardly useful for their speech. To overcome this problem, we organize novel and effective methods in the thesis.

In order to convert a speech style of a dysarthric person into that of like an unimpaired person, in Chapter 4, we presented a framework for parallel-data-free voice conversion (VC). Most of conventional VC systems require parallel data that aligned speech data from the source and the target speakers, which restricts the content of the utterance for training data. Such a system is hard to use for a person with an articulation disorder because the content that they can utter is limited owing athetoid symptoms. Non-negative matrix factorization (NMF) is a popular sparse representation and is applied for VC. However, NMF-VC also requires parallel data. To expand NMF-VC to a nonparallel method, we proposed a dictionary learning framework using non-negative Tucker decomposition (NTD). The dictionary is a weight matrix that represents a speaker identity and is made from parallel data. Our proposed NTD method decomposes an input spectrum into a speaker identity component, a shared component between speakers, and a speaker-dependent phonological component. This approach allows us to use not only parallel data but also any utterance as training data. In comparison

experiments, our proposed NTD-based dictionary learning method showed better performance than the traditional Gaussian mixture model-based method and comparable performance with the NMF-based method. Moreover, we also showed that our method is also effective to improve the intelligibility of dysarthric speech.

In Chapter 5, we presented an ent-to-end ASR (automatic speech recognition) system for persons with articulation disorders resulting from athetoid cerebral palsy. We proposed a transfer learning method using additional databases to solve the problem of limited data availability of their speech. Our method transfers two types of knowledge: the language-dependent characteristic of unimpaired speech and the language-independent characteristic of dysarthric speech. The former is transferred from speech data physically unimpaired Japanese-speaking persons, and the latter is transferred from speech data of English-speaking persons with articulation disorders. Because these additional speech data is obtained from publicly-available databases, we can achieve the well-trained model even when the amount of the target speaker's speech is limited. We demonstrated the effectiveness of the proposed approaches through the phoneme recognition task. The idea of using additional databases can be applied to the area where we do not have a sufficient amount of training data.

In Chapter 6, a multimodal ASR framework is introduced for a person with severe hearing loss. In the noisy environment, because the persons with hearing loss cannot hear ambient sounds, they cannot control the volumes of their voices and their speech style. Thus, the performance of ASR using only the speech signal significantly degrades. To compensate for the ASR performance, we utilize the lip movement in an utterance. Some people with hearing loss can communicate using lip reading and make the proper lip shape. Therefore, we proposed an audio-visual (multimodal) ASR for a person with hearing loss. Our method extracts the bottleneck feature from the convolutional neural networks that are trained to estimate the phoneme label. In comparison between the proposed feature and the conventional handcrafted feature, our proposed method showed better performances in the noisy environment. In the future, we will investigate the use of other modalities, e.g. the cheek and the throat.

To promote the symbiotic society of persons with and without disabilities, this thesis proposed new algorithms that help the social participation of persons with

disabilities. It is expected that our proposed technologies enable them to communicate with other people through his/her voice. Moreover, from an academic perspective, these approaches are not only related to assistive technology but also to typical-speech processing and other problems in the field of low-resource availability. Inspired by our work, we hope that related researches progress and persons with disabilities make advances into society.

References

[1] Frederic L. Darley, A.E. Aronson, and J.R. Brown. *Motor Speech Disorders*. Audio seminars in speech pathology. Saunders, 1975. 1

[2] Frank Rudzicz. Learning mixed acoustic/articulatory models for disabled speech. In *Proc. Neural Inf. Processing Syst.*, pages 70–78, Vancouver, BC, Canada, 2010. 1

[3] Eric Vatikiotis-Bateson, Inge-Marie Eigsti, Sumio Yano, and Kevin G. Munhall. Eye movement of perceivers during audiovisualspeech perception. *Perception & Psychophysics*, 60(6):926–940, Sep 1998. 2

[4] M. J. Tomlinson, M. J. Russell, and N. M. Brooke. Integrating audio and visual information to provide highly robust speech recognition. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 821–824, 1996. 2

[5] Ryo Aihara, Tetsuya Takiguchi, and Yasuo Ariki. Individuality-preserving voice conversion for articulation disorders using phoneme-categorized exemplars. *TACCESS*, 6(4):13:1–13:17, 2015. 2

[6] Yishan Jiao, Ming Tu, Visar Berisha, and Julie Liss. Simulating dysarthric speech for training data augmentation in clinical speech applications. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 6009–6013, Calgary, AB, Canada, 2018. 2, 54, 55

[7] M. Müller. *Information retrieval for music and motion*, volume 6. Springer, 2007. 4, 19

[8] Robert Gray. Vector quantization. *IEEE Assp Magazine*, 1(2):4–29, 1984. 4, 32

REFERENCES

[9] Hlne Valbret, Eric Moulines, and Jean-Pierre Tubach. Voice transformation using PSOLA technique. *Speech Communication*, 11(2-3):175–187, 1992. 4, 32

[10] Yannis Stylianou, Olivier Cappé, and Eric Moulines. Continuous probabilistic transform for voice conversion. *IEEE Trans. Speech and Audio Processing*, 6(2):131–142, 1998. 4, 32

[11] Tomoki Toda, Alan W. Black, and Keiichi Tokuda. Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory. *IEEE Trans. Audio, Speech & Language Processing*, 15(8):2222–2235, 2007. 4, 32, 39, 41

[12] Elina Helander, Tuomas Virtanen, Jani Nurminen, and Moncef Gabbouj. Voice conversion using partial least squares regression. *IEEE Trans. Audio, Speech & Language Processing*, 18(5):912–921, 2010. 4, 32

[13] Daisuke Saito, Hidenobu Doi, Nobuaki Minematsu, and Keikichi Hirose. Application of matrix variate gaussian mixture model to statistical voice conversion. In *Proc. Interspeech*, pages 2504–2508. ISCA, 2014. 4, 32

[14] Ryoichi Takashima, Tetsuya Takiguchi, and Yasuo Ariki. Exemplar-based voice conversion in noisy environment. In *IEEE Workshop on Spoken Language Technology*, pages 313–317, 2012. 4, 32

[15] Ryoichi Takashima, Ryo Aihara, Tetsuya Takiguchi, and Yasuo Ariki. Noise-robust voice conversion based on spectral mapping on sparse space. In *Speech Synthesis Workshop*, pages 71–75, 2013. 4, 32, 34, 39

[16] Srinivas Desai, E. Veera Raghavendra, B. Yegnanarayana, Alan W. Black, and Kishore Prahallad. Voice conversion using artificial neural networks. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pages 3893–3896, 2009. 4, 32

[17] Ling-Hui Chen, Zhen-Hua Ling, Yan Song, and Li-Rong Dai. Joint spectral distribution modeling using restricted Boltzmann machines for voice conversion. In *Proc. Interspeech*, pages 3052–3056, 2013. 4, 32

[18] Toru Nakashika, Tetsuya Takiguchi, and Yasuhiro Minami. Non-parallel training in voice conversion using an adaptive restricted Boltzmann machine. *IEEE/ACM Trans. Audio, Speech & Language Processing*, 24(11):2032–2045, 2016. 4, 24, 32, 39

[19] R. Aihara, T. Nakashika, T. Takiguchi, and Y. Ariki. Voice conversion based on non-negative matrix factorization using phoneme-categorized dictionary. *in Proc. ICASSP*, pages 7944–7948, 2014. 4

[20] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2000. 4, 32

[21] John S. Bridle, Michael D. Brown, and Richard M. Chamberlain. An algorithm for connected word recognition. In *ICASSP*, pages 899–902. IEEE, 1982. 5

[22] J. S. Bridle. Stochastic models and template matching: Some important relationships between two apparently different techniques for automatic speech recognition. *Proc. Inst, of Acoustics Autumn Conference*, 1984. 5, 27

[23] B. H. Juang. On the hidden markov model and dynamic time warping for speech recognition — A unified view. *AT&T Bell Laboratories Technical Journal*, 63(7):1213–1243, Sep. 1984. 5, 27

[24] M. A. Franzini, M. J. Witbrock, and K. . Lee. A connectionist approach to continuous speech recognition. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 425–428 vol.1, May 1989. 6

[25] Hervé Bourlard and Nelson Morgan. *Connectionist Speech Recognition: A Hybrid Approach*. Kluwer Press, 1994. 6, 28, 29

[26] A. Mohamed, G. Dahl, and G. Hinton. Deep belief networks for phone recognition. *NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, 1:9, 2009. 6, 52

[27] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *Proceedings of International Conference on Machine Learning*, pages 369–376. ACM, 2006. 6

[28] Hasim Sak, Andrew W. Senior, Kanishka Rao, Ozan Irsoy, Alex Graves, Franoise Beaufays, and Johan Schalkwyk. Learning acoustic frame labeling for speech recognition with recurrent neural networks. In *ICASSP*, pages 4280–4284. IEEE, 2015. 6, 29

[29] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R. Hershey, and Tomoki Hayashi. Hybrid ctc/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics Signal Processing*, 11(8):1240–1253, 2017. 6, 19

[30] C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani. State-of-the-art speech recognition with sequence-to-sequence models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778, April 2018. 6

[31] Ryo Aihara, Tetsuya Takiguchi, and Yasuo Ariki. Multiple non-negative matrix factorization for many-to-many voice conversion. *IEEE/ACM Trans. Audio, Speech & Language Processing*, 24(7):1175–1184, 2016. 9

[32] Berrak Sisman, Haizhou Li, and Kay Chen Tan. Sparse representation of phonetic features for voice conversion with and without parallel data. In *ASRU*, pages 677–684, Okinawa, Japan, 2017. IEEE. 9, 32, 33

[33] F. Fallside and W.A. Woods. *Computer speech processing*. Prentice-Hall International, 1985. 11

[34] L.R. Rabiner and R.W. Schafer. *Digital Processing of Speech Signals*. Prentice-Hall signal processing series. Prentice-Hall, 1978. 11

[35] B. P. Bogert, M. Healy, and J. W. Tukey. The quefrency alanysis of time series for echoes: Cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking. *in Proc. the symposium on time series analysis*, 15:209–243, 1963. 13

[36] P. Mermelstein. Distance measures for speech recognition, psychological and instrumental. *Pattern recognition and artificial intelligence*, 116:374–388, 1976. 14

[37] B. Milner and X. Shao. Speech reconstruction from mel-frequency cepstral coefficients using a source-filter model. *in Proc. Interspeech*, pages 2421–2424, 2002. 15

[38] Z. Tychtl and J. Psutka. Speech production based on the mel-frequency cepstral coefficients. *in Proc. EUROSPEECH*, 99:2335–2338, 1999. 16

[39] T. Ramabadran, J. Meunier, M. Jasiuk, and B. Kushner. Enhancing distributed speech recognition with back-end speech reconstruction. *in Proc. Interspeech*, pages 1859–1862, 2001. 16

[40] S. Imai. Cepstral analysis synthesis on the mel frequency scale. *in Proc. ICASSP*, 8:93–96, 1983. 16

[41] Masanori Morise, Fumiya Yokomori, and Kenji Ozawa. World: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Transactions*, 99-D(7):1877–1884, 2016. 16, 40

[42] H. Kawahara, I. Masuda-Katsuse, and A.de Cheveigne. Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequencybased F0 extraction: possible role of a repetitive structure in sounds. *Speech Communication*, 27(3-4):187–207, 1999. 16

[43] H. Kawahara. STRAIGHT, exploitation of the other aspect of vocoder: Perceptually isomorphic decomposition of speech sounds. *Acoustical Science and Technology*, 27(6):349–353, 2006. 16

[44] H. Kawahara, M. Morise, T. Takahashi, R. Nisimura, T. Irino, and H. Banno. TANDEM-STRAIGHT: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, f0, and aperiodicity estimation. *in Proc. ICASSP*, pages 3933–3936, 2008. 16

[45] D. Erro, I. Sainz, E. Navas, and I. Hernaez. Harmonics plus noise model based vocoder for statistical parametric speech synthesis. *IEEE J. Sel. Topics in Signal Process.*, 8(2):184–194, 2014. 17

[46] T. Nakashika. Voice conversion based on deep learning. *Doctral Thesis*, 2014. 19, 69

[47] R. Aihara. Voice conversion based on non-negative matrix factorization and its application to practical tasks. *Doctral Thesis*, 2017. 19

[48] Xuedong Huang, Yasuo Ariki, and Mervyn Jack. *Hidden Markov Models for Speech Recognition*. Columbia University Press, New York, NY, USA, 1990. 19

[49] Y. Stylianou, O. Cappe, and E. Moulines. Continuous probabilistic transform for voice conversion. *IEEE Trans. Speech and Audio Processing*, 6(2):131–142, 1998. 19

[50] A. R. Toth and A. W. Black. Using articulatory position data in voice transformation. *in Proc. ISCA SSW6*, pages 182–187, 2007. 19

[51] 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*, 39(1):1–38, 1977. 20

[52] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura. Speech parameter generation algorithms for hmm-based speech synthesis. *in Proc. ICASSP*, pages 1315–1318, 2000. 23

[53] T. Toda, A. Black, and K. Tokuda. Voice conversion based on maximum likelihood estimation of spectral parameter trajectory. *IEEE Trans. Audio, Speech, Lang. Process.*, 15(8):2222–2235, 2007. 23, 24

[54] S. Takamichi, T. Toda, A. W. Black, and S. Nakamura. Modulation spectrum-constrained trajectory training for gmm-based voice conversion. *in Proc. ICASSP*, pages 4859–4863, 2015. 24

[55] Yoav Freund and David Haussler. Unsupervised learning of distributions on binary vectors using two layer networks. Technical report, 1994. 24

[56] G E Hinton and R R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006. 24

[57] A. Llin K. Cho and T. Raiko. Improved learning of Gaussian-Bernoulli restricted Boltzmann machines. In *Artificial Neural Networks and Machine Learning*, pages 10–17, 2011. 24

[58] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006. 25

[59] Naoyuki Kanda, Xugang Lu, and Hisashi Kawai. Maximum a posteriori based decoding for ctc acoustic models. In *INTERSPEECH*, pages 1868–1872. ISCA, 2016. 29

[60] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *ICML*, pages 1764–1772, 2014. 29

[61] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *Proc. Neural Inf. Processing Syst.*, pages 577–585, Montreal, Quebec, Canada, 2015. 29, 56

[62] Yuki Takashima, Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki. Non-parallel dictionary learning for voice conversion using non-negative tucker decomposition. *EURASIP Journal on Audio, Speech, and Music Processing*, 2019(1):17, Sep 2019. 31

[63] Tomoki Toda, Ling-Hui Chen, Daisuke Saito, Fernando Villavicencio, Mirjam Wester, Zhizheng Wu, and Junichi Yamagishi. The voice conversion challenge 2016. In *Proc. Interspeech*, pages 1632–1636, San Francisco, California, USA, 2016. ISCA. 32

[64] Alexander Kain and Michael W. Macon. Spectral voice conversion for text-to-speech synthesis. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pages 285–288, Seattle, Washington, USA, 1998. IEEE. 32

[65] Christophe Veaux and Xavier Rodet. Intonation conversion from neutral to expressive speech. In *Proc. Interspeech*, pages 2765–2768, Florence, Italy, 2011. ISCA. 32

[66] Keigo Nakamura, Tomoki Toda, Hiroshi Saruwatari, and Kiyohiro Shikano. Speaking-aid systems using GMM-based voice conversion for electrolaryngeal speech. *Speech Communication*, 54(1):134–146, 2012. 32

[67] Li Deng, Alex Acero, Li Jiang, Jasha Droppo, and Xuedong Huang. High-performance robust speech recognition using stereo training data. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pages 301–304, Salt Lake City, Utah, USA, 2001. IEEE. 32

[68] Aki Kunikoshi, Yu Qiao, Nobuaki Minematsu, and Keikichi Hirose. Speech generation from hand gestures based on space mapping. In *Proc. Interspeech*, pages 308–311, Brighton, United Kingdom, 2009. ISCA. 32

[69] Zhizheng Wu, Tuomas Virtanen, Engsiong Chng, and Haizhou Li. Exemplar-based sparse representation with residual compensation for voice conversion. *IEEE/ACM Trans. Audio, Speech & Language Processing*, 22(10):1506–1521, 2014. 32

[70] Toru Nakashika, Ryoichi Takashima, Tetsuya Takiguchi, and Yasuo Ariki. Voice conversion in high-order eigen space using deep belief nets. In *Proc. Interspeech*, pages 369–372, Lyon, France, 2013. ISCA. 32

[71] Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki. Voice conversion using rnn pre-trained by recurrent temporal restricted Boltzmann machines. *IEEE/ACM Trans. Audio, Speech & Language Processing*, 23(3):580–587, 2015. 32

[72] Zhizheng Wu, Engsiong Chng, and Haizhou Li. Conditional restricted Boltzmann machine for voice conversion. In *ChinaSIP*, pages 104–108, Beijing, China, 2013. IEEE. 32

[73] Yuki Saito, Yusuke Ijima, Kyosuke Nishida, and Shinnosuke Takamichi. Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and d-vectors. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pages 5274–5278, Calgary, AB, Canada, 2018. IEEE. 32

[74] Fuming Fang, Junichi Yamagishi, Isao Echizen, and Jaime Lorenzo-Trueba. High-quality nonparallel voice conversion based on cycle-consistent adversarial network. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pages 5279–5283, Calgary, AB, Canada, 2018. IEEE. 32

[75] Ryo Aihara, Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki. Voice conversion based on non-negative matrix factorization using phoneme-categorized dictionary. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pages 7894–7898, Florence, Italy, 2014. IEEE. 32, 36

[76] Ryo Aihara, Tetsuya Takiguchi, and Yasuo Ariki. Parallel dictionary learning for voice conversion using discriminative graph-embedded non-negative matrix factorization. In *Proc. Interspeech*, pages 292–296, San Francisco, CA, USA, 2016. ISCA. 32, 36

[77] Ryo Aihara, Tetsuya Takiguchi, and Yasuo Ariki. Activity-mapping non-negative matrix factorization for exemplar-based voice conversion. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pages 4899–4903, South Brisbane, Queensland, Australia, 2015. IEEE. 32, 36

[78] Athanasios Mouchtaris, Jan Van der Spiegel, and Paul Mueller. Nonparallel training for voice conversion based on a parameter adaptation approach. *IEEE Trans. Audio, Speech & Language Processing*, 14(3):952–963, 2006. 32

[79] Tetsuya Hashimoto, Hidetsugu Uchida, Daisuke Saito, and Nobuaki Minematsu. Parallel-data-free many-to-many voice conversion based on dnn integrated with eigenspace using a non-parallel speech corpus. In *Proc. Interspeech*, pages 1278–1282, Stockholm, Sweden, 2017. ISCA. 32

[80] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4):1253–1278, 2000. 33

[81] P. M. Kroonenberg and J. De Leeuw. Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika*, 45:69–97, 1980. 33

[82] L. R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311, 1966. 33

[83] Hedi Ben younes, Rmi Cadne, Matthieu Cord, and Nicolas Thome. Mutan: Multimodal tucker fusion for visual question answering. In *ICCV*, pages 2631–2639, Venice, Italy, 2017. IEEE Computer Society. 33

[84] Jen-Tzung Chien and Chen Shen. Deep neural factorization for speech recognition. In *Proc. Interspeech*, pages 3682–3686, 2017. 33

[85] Daisuke Saito, Keisuke Yamamoto, Nobuaki Minematsu, and Keikichi Hirose. One-to-many voice conversion based on tensor representation of speaker space. In *Proc. Interspeech*, pages 653–656, Florence, Italy, 2011. ISCA. 33

[86] Zhanyu Ma, Andrew E. Teschendorff, Arne Leijon, Yuanyuan Qiao, Honggang Zhang, and Jun Guo. Variational bayesian matrix factorization for bounded support data. *IEEE transactions on pattern analysis and machine intelligence*, 37(4):876–889, 2015. 33

[87] Qiquan Shi, Yiu-Ming Cheung, Qibin Zhao, and Haiping Lu. Feature extraction for incomplete data via low-rank tensor decomposition with feature regularization. *IEEE Transactions on Neural Networks and Learning Systems*, 30(6):1803–1817, 2019. 33

[88] Bo Jiang, Chris Ding, Jin Tang, and Bin Luo. Image representation and learning with graph-laplacian Tucker tensor decomposition. *IEEE Transactions on Cybernetics*, 49(4):1417–1426, 2019. 33

[89] Y. Kim and S. Choi. Nonnegative tucker decomposition. In *Computer Vision and Pattern Recognition*, pages 1–8, Minneapolis, Minnesota, USA, 2007. IEEE Computer Society. 36, 37

[90] Akira Kurematsu, Kazuya Takeda, Yoshinori Sagisaka, Shigeru Katagiri, Hisao Kuwabara, and Kiyohiro Shikano. ATR Japanese speech database as a tool of speech recognition and synthesis. *Speech Communication*, 9(4):357–363, 1990. 39, 62

[91] Ryo Aihara, Takao Fujii, Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki. Small-parallel exemplar-based voice conversion in noisy environments using affine non-negative matrix factorization. *EURASIP J. Audio, Speech and Music Processing*, 2015:32, 2015. 40

[92] Kazuhiro Kobayashi and Tomoki Toda. sprocket: Open-source voice conversion software. In *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, pages 203–210, Les Sables d'Olonne, France, 2018. ISCA. 40

[93] Jaime Lorenzo-Trueba, Junichi Yamagishi, Tomoki Toda, Daisuke Saito, Fernando Villavicencio, Tomi Kinnunen, and Zhen-Hua Ling. The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods. In *Odyssey*, pages 195–202, Les Sables d'Olonne, France, 2018. ISCA. 47

[94] Ryo Aihara, Ryoichi Takashima, Tetsuya Takiguchi, and Yasuo Ariki. Noise-robust voice conversion based on sparse spectral mapping using non-negative matrix factorization. *IEICE Transactions on Information and Systems*, 97-D(6):1411–1418, 2014. 50

[95] Yuki Takashima, Ryoichi Takashima, Tetsuya Takiguchi, and Yasuo Ariki. Knowledge transferability between the speech data of persons with dysarthria speaking different languages for dysarthric speech recognition. *IEEE Access*, 2019. 51

[96] George E. Dahl, Dong Yu, Li Deng, and Alex Acero. Large vocabulary continuous speech recognition with context-dependent DBN-HMMs. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 4688–4691, Prague, Czech Republic, 2011. 52

[97] Tara N. Sainath, Abdel rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran. Deep convolutional neural networks for LVCSR. In *Proc. IEEE*

Int. Conf. Acoust., Speech, Signal Processing, pages 8614–8618, Vancouver, BC, Canada, 2013. 52

[98] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. on Knowl. Data Eng.*, 22(10):1345–1359, 2010. 53

[99] Yuki Takashima, Tetsuya Takiguchi, and Yasuo Ariki. End-to-end dysarthric speech recognition using multiple databases. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 6395–6399, Brighton, United Kingdom, 2019. 53, 56, 59

[100] Xavier Menéndez-Pidal, James B. Polikoff, Shirley M. Peters, Jennie E. Leonzio, and H. Timothy Bunnell. The nemours database of dysarthric speech. In *Proc. 4th Int. Conf. Spoken Lang. Processing*, Philadelphia, PA, USA, 1996. 54, 55

[101] Heejin Kim, Mark Hasegawa-Johnson, Adrienne Perlman, Jon Gunderson, Thomas S. Huang, Kenneth Watkin, and Simone Frame. Dysarthric speech database for universal access research. In *Proc. INTERSPEECH*, pages 1741–1744, Brisbane, Australia, 2008. 54, 55

[102] Frank Rudzicz, Aravind Kumar Namasivayam, and Talya Wolff. The TORGO database of acoustic and articulatory speech from speakers with dysarthria. *Language Resources and Evaluation*, 46(4):523–541, 2012. 54, 55, 62

[103] D. Choi, B. Kim, Y. Lee, Y. Um, and M. Chung. Design and creation of dysarthric speech database for development of QoLT software technology. In *International Conference on Speech Database and Assessments (Oriental COCOSDA)*, pages 47–50, 2011. 54

[104] Ka-Ho Wong, Yu Ting Yeung, Edwin H. Y. Chan, Patrick C. M. Wong, Gina-Anne Levow, and Helen M. Meng. Development of a Cantonese dysarthric speech corpus. In *Proc. INTERSPEECH*, pages 329–333, Dresden, Germany, 2015. 54

[105] Bhavik Vachhani, Chitralekha Bhat, Biswajit Das, and Sunil Kumar Kopparapu. Deep autoencoder based speech features for improved dysarthric speech recognition. In *Proc. INTERSPEECH*, pages 1854–1858, Stockholm, Sweden, 2017. 54, 55

[106] Feifei Xiong, Jon Barker, and Heidi Christensen. Phonetic analysis of dysarthric speech tempo and applications to robust personalised dysarthric speech recognition. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 5836–5840, Brighton, United Kingdom, 2019. 54

[107] Hironori Matsumasa, Tetsuya Takiguchi, Yasuo Ariki, Ichao Li, and Toshitaka Nakabayashi. Integration of metamodel and acoustic model for speech recognition. In *Proc. INTERSPEECH*, pages 2234–2237, Brisbane, Australia, 2008. 54

[108] Chikoto Miyamoto, Yuto Komai, Tetsuya Takiguchi, Yasuo Ariki, and Ichao Li. Multimodal speech recognition of a person with articulation disorders using AAM and MAF. In *Proc. IEEE Int. Workshop Multimedia Signal Processing*, pages 517–520, Saint Malo, France, 2010. 54, 69

[109] Yuki Takashima, Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki. Feature extraction using pre-trained convolutive bottleneck nets for dysarthric speech recognition. In *Proc. EUSIPCO*, pages 1411–1415, Nice, France, 2015. 54

[110] Ryo Aihara, Tetsuya Takiguchi, and Yasuo Ariki. Phoneme-discriminative features for dysarthric speech conversion. In *Proc. INTERSPEECH*, pages 3374–3378, Stockholm, Sweden, 2017. 54

[111] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. Neural Inf. Processing Syst.*, pages 2672–2680, Montreal, Quebec, Canada, 2014. 55

[112] Bhavik Vachhani, Chitralekha Bhat, and Sunil Kumar Kopparapu. Data augmentation using healthy speech for dysarthric speech recognition. In *Proc. INTERSPEECH*, pages 471–475, Hyderabad, India, 2018. 55

[113] Neethu Mariam Joy, S. Umesh, and Basil Abraham. On improving acoustic models for TORGO dysarthric speech database. In *Proc. INTERSPEECH*, pages 2695–2699, Stockholm, Sweden, 2017. 55, 65

[114] S. Chandrakala and N. Rajeswari. Representation learning based speech assistive system for persons with dysarthria. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 25(9):1510–1517, 2017. 55

[115] N. M. Joy and S. Umesh. Improving acoustic models in TORGO dysarthric speech database. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 26(3):637–645, 2018. 55

[116] Shubham Toshniwal, Tara N. Sainath, Ron J. Weiss, Bo Li, Pedro J. Moreno, Eugene Weinstein, and Kanishka Rao. Multilingual speech recognition with a single end-to-end model. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 4904–4908, Calgary, AB, Canada, 2018. 55, 64

[117] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proc. Int. Conf. Machine Learning*, Pittsburgh, Pennsylvania, USA, 2006. 56

[118] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 4960–4964, Shanghai, China, 2016. 56

[119] Rich Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997. 60

[120] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 62

[121] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 2818–2826, Las Vegas, NV, USA, 2016. 63

[122] Yuki Takashima, Yasuhiro Kakihara, Ryo Aihara, Tetsuya Takiguchi, Yasuo Ariki, Nobuyuki Mitani, Kiyohiro Omori, and Kaoru Nakazono. Audio-visual speech recognition using convolutive bottleneck networks for a person with severe hearing loss. *IPSJ Trans. Computer Vision and Applications*, 7:64–68, 2015. 67

[123] J. Lin, W. Ying, and T. S. Huang. Capturing human hand motion in image sequences. *in Proc. IEEE Motion and Video Computing Workshop*, pages 99–104, 2002. 68

[124] T. Starner, J. Weaver, and A. Pentland. Real-time American sign language recognition using desk and wearable computer based video. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, 1998. 68

[125] N. Ezaki, M. Bulacu, and L. Schomaker. Text detection from natural scene images: Towards a system for visually impaired persons. *in Proc. Int. Conf. on Pattern Recognition*, pages 683–686, 2004. 68

[126] H. Matsumasa, T. Takiguchi, Y. Ariki, I. Li, and T. Nakabayashi. Integration of metamodel and acoustic model for dysarthric speech recognition. *Journal of Multimedia*, 4(4):254–261, 2009. 68, 69

[127] G. Potamianos and H. P. Graf. Discriminative training of HMM stream exponents for audio-visual speech recognition. *in Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pages 3733–3736, 1998. 68, 69

[128] M. J. Tomlinson, M. J. Russell, and N. M. Brooke. Integrating audio and visual information to provide highly robust speech recognition. *in Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pages 821–824, 1996. 68, 69

[129] A. Verma, T. Faruquie, C. Neti, S. Basu, and A. Senior. Late integration in audio-visual continuous speech recognition. *in Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, 1999. 68, 69

[130] K. Vesely, M. Karafiat, and F. Grezl. Convolutive bottleneck network features for lvcsr. *in Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 42–47, 2011. 68

[131] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *in Proc. the IEEE*, 86(11):2278–2324, 1998. 68, 69

[132] Honglak Lee, Yan Largman, Peter Pham, and Andrew Y. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. *in Proc. Neural Information Processing Systems*, 22:1096–1104, 2009. 68, 69

[133] T. F. Cootes. Active appearance models. *in Proc. European Conf. on Computer Vision*, 2:484–498, 1998. 69

[134] K.L. Sum, WH. Lau, S.H. Leung, A. W. C. Liew, and K. W. Tse. A new optimization procedure for extracting the point-based lip contour using active shape

model. *in Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pages 1485–1488, 2001. 69

[135] David Cristinacce and Timothy F. Cootes. Feature detection and tracking with constrained local models. *in Proc. British Machine Vision Conf.*, 2(5):929–938, 2006. 69

[136] Jason M. Saragih, Simon Lucey, and Jeffrey F. Cohn. Deformable model fitting by regularized landmark mean-shift. *Int. Journal of Computer Vision*, 91(2):200–215, 2011. 69

[137] G. Hinton, Deng Li, Yu Dong, G.E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82 – 97, 2012. 69

[138] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A.Y. Ng. Multimodal deep learning. *in Proc. International Conference on Machine Learning*, 2011. 69

[139] M. Delakis and C. Garcia. Text detection with convolutional neural networks. *in Proc. Int. Conf. on Computer Vision Theory and Applications*, pages 290–294, 2008. 69

[140] G. Montavon. Deep learning for spoken language identification. *in Proc. Workshop on Deep Learning for NIPS*, 2009. 69

Acknowledgements

First, I would like to thank my supervisors, Emeritus Professor Yasuo Ariki, Professor Tetsuya Takiguchi, and Associate Professor Ryoichi Takashima at Kobe University, Assistant Professor Toru Nakashika at The University of Electro-Communications, and Ryo Aihara at Mitsubishi Electric Corporation, who have given me helpful advice and continued support during my research and writing up. Their broad knowledge in the field and his down-to-earth attitude has been of great help to my study. I also thank Professor Sano and Professor Tamaki for their constructive comments and valuable suggestions, which helped to improve this thesis.

Meanwhile, I would like to thank the past and the present members in CS 17 Media Lab., where we have done efforts together and shared joys and sorrows of research life.

Finally, to my family and my friends, it cannot be described in words, yet I would like to show my full gratitude for their love, encouragements, and unconditional support throughout my study.

BibTeX Citation for This Thesis

```
@article  {Y. TakashimaKBUPhDThesis2020,
           title={{Assistive Technology Using Machine Learning
                   Based on Multi-Domain Data}},
                   for Articulation Disorders}},
           journal={Doctoral Thesis},
           author={Yuki Takashima},
           institution={Kobe University},
           month={Mar.},
           year={2020}
          }
```

Doctor Thesis, Kobe University
"Assistive Technology Using Machine Learning Based on Multi-Domain Data for Articulation Disorders", 124 pages
Submitted on January, 23, 2020.
The date of publication is printed in the cover of repository version published in Kobe University Repository Kernel.