# A Study on Environmental Sound Modeling based on Deep Learning

Deparment of Media Science
Graduate School of Information Science
Nagoya University

Tomoki Hayashi

# Contents

# Abstract

With recent increases in computational power, environmental sound understanding has become more feasible. Researchers intend to develop automated systems that can identify every possible sound in a given environment, from the sound of glass breaking to the crying of children. One of the most important tasks in this field is sound event detection (SED), which is the task of detecting the beginning and the end of sound events and labeling them. Sound events include a wide range of phenomena that vary widely in acoustic characteristics, duration, and volume, such as the sound of glass breaking, typing on a keyboard, knocking on doors, and human speech. This diversity of targets makes SED challenging. Though recent advances in machine learning techniques have improved the performance of SED systems, various problems remain to be solved.

This thesis addresses three problems that affect the performance of monophonic, polyphonic, and anomalous SED systems. The first is how to different types of signals can be used in combination, to extend the range of detectable sound events. The second is how to model the duration of sound events, which is among the most important characteristics, to improve polyphonic SED performance. The third is how to model normal environments, in which no anomalous sound events occur, to improve the performance of anomalous SED systems.

As a part of the work in developing a life-logging system, this work focuses on the use of multi-modal signals to extend the range of detectable sound events into the area of common human activities. The key to realizing these applications is finding ways to associate different types of signals to detect variety of human activities. First two deep neural network (DNN)-based fusion methods using multi-modal signals are proposed for this associative goal. Then, a large database of human activities recorded under realistic conditions is created for testing the performance of the proposed methods. Furthermore, to address the problem of model individuality, which degrades system performance, speaker adaptation techniques from the field of automatic speech recognition are introduced. Experimental results using the constructed database demonstrate that the use of multi-modal signals is effective, and that speaker adaptation techniques can improve performance, especially when using only a limited amount of training data.

To improve the performance of polyphonic SED systems, this work focuses on modeling the duration of sound events. To do this, a novel hybrid approach using duration-controlled long short-term memory (LSTM) is proposed, which builds upon a state-of-the-art SED method that performs frame-by-frame detection using a bidirectional LSTM recurrent neural network (BLSTM) by incorporating a duration-controlled modeling technique based on a hidden Markov model (HMM) or a hidden semi-Markov model (HSMM). The proposed approach makes it possible to model the duration of each sound event precisely and to perform sequence-by-sequence detection without needing thresholding. Furthermore, to effectively reduce insertion errors, post-processing method using binary masks is also introduced. This post-processing step uses a sound activity detection (SAD) network to identify segments for activity indicating any sound event. Experimental evaluation with the DCASE2016 task2 dataset demonstrates that the proposed method outperforms conventional polyphonic SED methods, proving that sound event duration is effectively modeled for polyphonic SED.

The key to successful anomalous SED is in finding a method for modeling the normal acoustic environment. This modeling is performed conventionally in the acoustic feature domain, but this results in a lack of information about temporal structure like the phase of the sounds. To address this issue, a new anomalous detection method based on WaveNet, which is an autoregressive convolutional neural network that directly models acoustic signals in the time domain, is proposed. The proposed method uses WaveNet as a predictor rather than as a generator to detect waveform segments that are responsible for large prediction errors as unknown acoustic patterns. Furthermore, to consider differences in environmental situations, i-vector is utilized as an additional auxiliary feature of WaveNet. The i-vector extractor should allow the system to discriminate the sound patterns, depending on the time, location, and surrounding environment. Experimental evaluation with a database of sounds recorded in public spaces shows that the proposed method outperforms conventional feature-based approaches, and that time-domain modeling in conjunction with the i-vector extractor is effective for anomalous SED.

# 1 Introduction

## 1.1 Background

Humans encounter many kinds of sounds in daily life, such as speech, music, the singing of birds, and keyboard typing. Over the past several decades, the main targets of acoustic research have been speech and music, while other sounds have generally been treated as background noise. Recently, with the improvement of machine learning techniques, automated systems can identify a wider range of sounds in an environment. Understanding environmental sound is challenging because if the goal is to understand every possible sound in a given environment. This task includes acoustic scene classification, sound event detection, and audio tagging. Several contests have been held in recent years, including CLEAR AED [1], TRECVID MED [2], and the DCASE Challenge [3–6]. Moreover, several datasets have been developed for the purpose of such research, such as urban sound datasets [7], and AudioSet [8].

One of the most exciting tasks in this field is sound event detection (SED), which involves the detection of the beginning and the end of sound events, along with indetifying their type. SED has many applications such as retrieval from multimedia databases [9], life-logging [10, 11], automatic control of devices in smart homes [12], and surveillance systems [13–15].

SED tasks can be divided into three categories, monophonic, polyphonic, and anomalous, depending on the scenario represented by the environment. In monophonic SED, the types of sound events to be detected are predefined and the maximum number of simultaneously active sound events is assumed to be one. Since sound events include a wide range of phenomena, which vary widely in their acoustic characteristics, duration, and volume, correctly detecting and identifying these sound events is a challenging task even if in the case of monophonic scenarios.

On the other hand, polyphonic SED involves any number of simultaneously active sound events. Polyphonic SED is more realistic than monophonic SED because several sound events are likely to co-occur in any real environment. This is even more difficult than monophonic SED, due to the overlapping of multiple sound events.

In contrast to monophonic and polyphonic SED, anomalous SED is to detect sound events that stand out from a particular context, such as the sound of screaming or gunshots in public spaces. Since anomalous sound event data is difficult to collect, anomalous SED systems are generally designed to function in an unsupervised manner.

Each of the above SED techniques can be applied in various scenarios, and each has great potential in various applications. However, the current level of the performance is still insufficient for practical applications. There is still much room for improvement, so each type of scenario requires a range of problems to be addressed to achieve acceptable performance.

## 1.2 Scope of this Thesis

This thesis addresses three problems that limit the performance of monophonic, polyphonic, and anomalous SED systems. The first is how to combine different kinds of signals from wearable sensors in order to extend the range of detectable sound events. The second is how to model the duration of sound events to improve polyphonic SED performance. And the third is how to model normal environments, in which no anomalous sound events occur, to improve the performance of anomalous SED systems. In this section, each problem is briefly introduced.

### 1.2.1 Human activity recognition based on deep neural network using multi-modal signals

Toward the development of a life-logging system, this work focuses on the use of multi-modal signals recorded under realistic conditions so that sound events related to common human activities can be detected, including discrete sound events like door closing along with sounds related to more-extended human activities like cooking. The key to realizing these applications is finding associations between different types of signals that facilitate the detection of various human activities. Below, two deep neural network (DNN)-based fusion methods are proposed to join with multi-modal signals together. Then, a large-scale database of human activities recorded in the real world is created for testing the performance of the proposed methods. Furthermore, to address the problem of model individuality, speaker adaptation techniques used in the field of automatic speech recognition are introduced. Experimental evaluation with the constructed database demonstrates that the use of multi-modal signals is effective, and that speaker adaptation techniques can improve performance, especially when only a limited amount of training data is used.

### 1.2.2 Polyphonic SED based on duration modeling

In order to improve polyphonic SED performance, this work focuses on modeling the duration of sound events, which is one of the most important characteristics that distinguishes different kinds of events. A new hybrid approach using duration-controlled long short-term memory (LSTM) is proposed, which builds upon a state-of-the-art SED method that performs frame-by-frame detection using a bidirectional LSTM recurrent neural network (BLSTM). This method incorporates a duration-controlled modeling technique based on a hidden Markov model (HMM) or a hidden semi-Markov model (HSMM). The proposed approach makes it possible to model the duration of each sound event precisely and to perform sequence-by-sequence detection without the need for thresholding, as is required in conventional frame-by-frame methods. Furthermore, to effectively reduce insertion errors, which often occur under noisy conditions, a post-processing method based on binary masks is also introduced, which relies on a sound activity detection (SAD) network to identify segments with any sound event activity. Experimental evaluation with the DCASE2016 task2 dataset demonstrates that the proposed method outperforms conventional polyphonic SED methods, suggesting that modeling sound event duration is effective for polyphonic SED.

### 1.2.3 Anomalous SED based on direct waveform modeling

This work focuses on the problem of modeling normal acoustic environments to facilitate anomalous SED. Such modeling is performed conventionally in the acoustic feature domain because acoustic signals have high sampling rates and non-stationarity that make them hard to treat directly in the time domain. However, this modeling approach excludes temporal information like the phase of the signal. Therefore, if such structures can be modeled in the time domain, detection performance should be improved. To do this, a new anomalous SED method is proposed based on WaveNet [16]. WaveNet is a generative model based on an auto-regressive convolutional neural network (CNN) that can generate human speech using random sampling. The proposed method uses WaveNet as a predictor rather than as a generator to detect the waveform segments responsible for the large prediction errors as unknown acoustic patterns. Furthermore, to take differences in environmental situations into consideration, i-vector is used as an additional auxiliary feature of WaveNet, which has been utilized in the field of speaker verification. This i-vector extractor will enable discrimination of the sound patterns, depending on the time, location, and surrounding environment. Experimental evaluation using a database recorded in public spaces demonstrate that the proposed method outperforms conventional feature-based approaches, and that modeling in the time domain in conjunction with the i-vector extractor is effective for anomalous SED.

## 1.3 Relationship with RWDC

For the success of a commercial application, real-world data must be collected and be utilized continuously. This process includes feedback from end users along with the analysis and application of that data for new products and services. This process of circulating data from acquisition, to analysis, and to implementation is known as real-world data circulation (RWDC) [17]. This thesis introduces the concept of RWDC and discusses each work in terms of RWDC. This thesis also describes that how the developed tool for human activity recognition can be used to foster RWDC, how the analysis of data during research on polyphonic SED inspires a new method based on duration modeling, and how outputs from the research on anomalous SED can be used to improve the performance of polyphonic SED systems.

## 1.4 Thesis Overview

The thesis is organized as follows. In Chapter 2, environmental sound understanding and its applications are discussed. In Chapter 3, a human activity recognition based on a deep neural network using multi-modal signals is described. Chapter 4 discusses polyphonic SED based on a duration controlled model. In Chapter 5, a method for anomalous SED based on waveform modeling is proposed. In Chapter 6, the relationship with RWDC is discussed. Finally, in Chapter 7, the contributions of this thesis are summarized and future work is discussed.

# 2 Environmental Sound Understanding and its Applications

The field of environmental sound understanding includes various research topics that have great potential across a range of applications. This chapter focuses on the four main tasks in environmental sound understanding, describing the definition of each problem and each task's applications. The relationship between these tasks is then discussed in terms of two criteria: the length of the reference time and the degree of identification. Moreover, various techniques used to achieve each task are reviewed.

## 2.1 Introduction

Humans are always surrounded by a variety of sounds, such as speech, music, the chirping of birds, keyboard typing, and traffic noise. Even when these sounds occur simultaneously, human beings can select which sound to pay attention and can usually identify it and determine where it is coming from. This phenomenon is called the cocktail party effect [18–20]. Many researchers have studied this human ability and have attempted to replicate it with computational machines, a field known as computational auditory scene analysis (CASA), and various techniques have been proposed since the 1990s [21–25]. These studies have mainly focused on speech or music, but not other types of sounds, and therefore, the scope of CASA's applications is limited.

With the development of more powerful machine learning techniques and the availability of increasing computational power, the field of environmental sound understanding has emerged, the goal of which is to understand the whole range of environmental sounds. The wide range of target sounds means that many applications are possible, such as life-logging, multimedia retrieval, indexing, and surveillance. Several challenges have been held for environmental sound understanding, such as CLEAR AED [1], TRECVID MED [2], and the DCASE Challenge [3–6]. Furthermore, several datasets have been developed, including urban sound datasets [7] and AudioSet [8].

Typical tasks in the field of environmental sound understanding include acoustic scene classification (ASC), sound event detection (SED), audio tagging, and anomalous sound event detection (anomalous SED). According to Oishi [26], these tasks can be categorized using two axes: the length of the reference time and the degree of identification. These axes are shown in Fig. 2.1, where the length of the reference time represents the amount of time that can be attributed to recognized targets, and the degree of identification represents the precision with which the target sound is specified. If the reference time is short, the system can detect or recognize sounds within a short frame of time, such as during voice activity detection (VAD) [27]. If the length of the reference time is long, this means that the system identifies sounds based on the characteristics of a longer signal, such as during speaker recognition [28]. On the other hand, if the degree of identification is high, the system recognizes sounds that are an exact match to the reference sound. If the degree of identification is low, the system simply identifies sounds that share characteristics in common with target sound events.

This chapter introduces the definitions and techniques associated with each task, as well as the applications for these tasks. Sections 2.2, 2.3, 2.4, and 2.5 describe the problem definitions, techniques and applications of ASC, SED, audio tagging and anomalous SED,

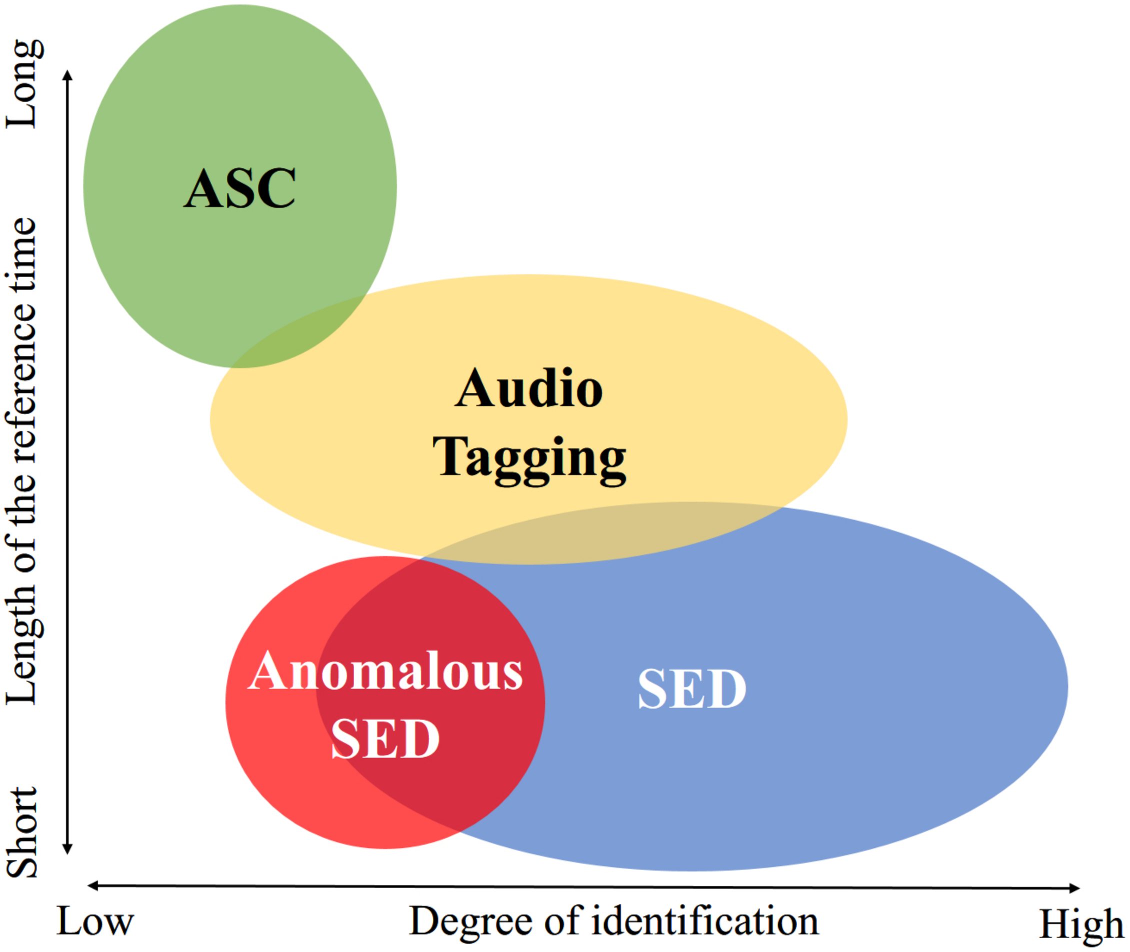


Figure 2.1: *Relative positions of each task in the field of environmental sound understanding.*

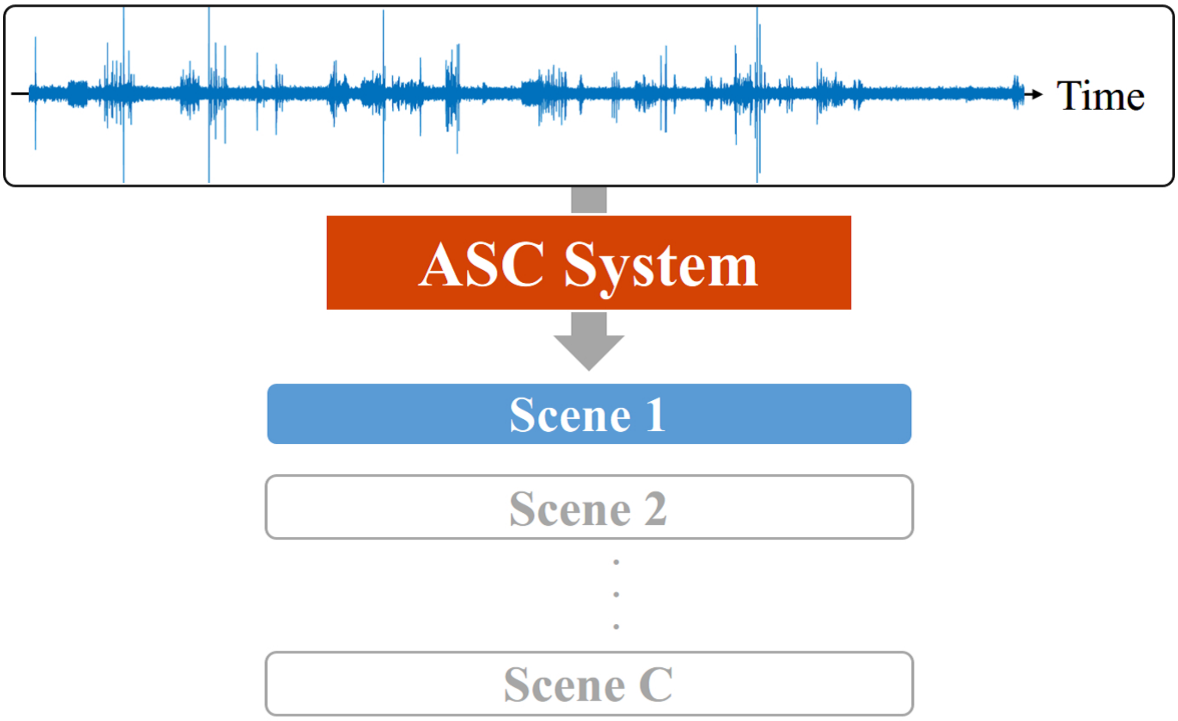


Figure 2.2: *Overview of an acoustic scene classification system.*

respectively. Finally, this chapter is summarized in Section 2.6.

## 2.2 Acoustic Scene Classification

ASC classifies a recorded signal into one of several predefined classes that describe the environment in which the signal was recorded, such as an office, a beach, or a cafe. An overview of a typical ASC system is shown in Fig. 2.2. ASC techniques can be applied for indexing media content for retrieval [9] and automatic mode-switching [29] for context-aware robots [30]. Even if the location where the signal was recorded is the same, the components included in the recorded signal can vary widely, e.g., a quiet day at the office versus a very busy day at the office. Therefore, the degree of identification for this task is low and it is necessary to use statistics of relatively long-term signals to represent the global characteristics of the environment. In this respect, ASC is related to music genre classification [31, 32] and speaker recognition [28, 33]. However, fixed-length signals (e.g. 30 seconds) are often prepared as training data for ASC, but speaker recognition systems must deal with data segments that contain speech of various lengths.

One of the most used approaches to ASC is the bag-of-words algorithm [34, 35]. This approach extracts various fixed-dimension features such as the Mel frequency cepstrum coefficients (MFCCs) from each frame, and then uses them to create a histogram using a codebook created by vector quantization. Finally, a classifier, such as support vector machine (SVM), or a generative model, such as a Gaussian mixture model (GMM), is trained using the histogram as the input. Another common approach is majority voting [36], which performs frame-wise classification, and then tabulates the results to decide the final output class.

Inspired by studies in the field of speaker recognition [33], i-vector extraction has also been utilized for ASC [37, 38]. Since i-vector contains information related to speaker characteristics and recording conditions, as in the case of speaker recognition, it is expected that i-vectors can also capture the global characteristics of acoustic scenes.

More recently, inspired by the success achieved in the field of image recognition using convolutional neural networks (CNN) [39–42], many researchers have utilized CNN for ASC with promising results [43–45]. Since sequences of acoustic features can be rendered as 2D images, CNN models can be applied to ASC in the same manner as they are used for image recognition. While previous ASC studies utilized hand-crafted features such as MFCCs, studies using CNNs have generally focused on low-level features such as spectrogram or log Mel-filter banks, due to the exceptional performance of CNNs as feature extractors. Some studies have returned accurate classifications even when raw waveforms are used directly as the input [46, 47]. CNN approaches require a large amount of training data to achieve accu-

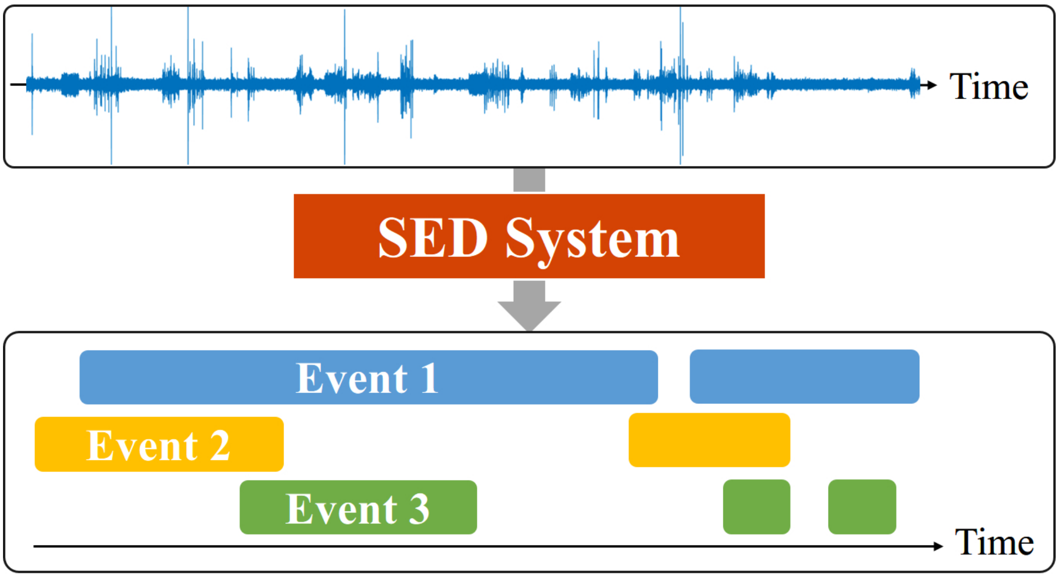


Figure 2.3: *Overview of an sound event detection system.*

rate classification. Thus, the development of data augmentation methods that can increase the volume of training data artificially has become a focus of research in this field [48, 49].

## 2.3 Sound Event Detection

SED is the task of detecting the start and the end time and labeling predefined sound events. An overview of an SED system is shown in Fig. 2.3. The main difference between ASC and SED is that SED systems attempt to classify sound events and to detect their start and end times. Furthermore, while the classes used for ASC are mutually exclusive, SED can be used to identify overlapping sound events. Moreover, SED is generally focused on the identification of particular sound events, rather than the identification broader acoustic environments. SED has many applications, such as retrieval from multimedia databases [9], life-logging [10, 11], automatic control of devices in smart homes [12], and surveillance systems [13–15]. The degree of identification depends on the user's definition of what represents a sound event, but the length of the reference time tends to be short because the system detects single-sound events within long acoustic signals. Therefore, local and global features are relevant to SED.

SED can be divided into two basic types: monophonic and polyphonic. For monophonic SED, sound events are predefined and the maximum number of simultaneously active sound events is assumed to be one. Since sound events include a wide range of sounds, that vary in acoustic characteristics, duration, and volume, correctly detecting and identifying them is a challenging task, even in the case of monophonic SED. In polyphonic SED, there can be any number of simultaneously active sound events. Polyphonic SED is a more realistic task than monophonic SED because several sound events usually occur simultaneously in real-world situations, but polyphonic SED is much more difficult than monophonic SED.

The most typical method for SED is a hidden Markov model (HMM), where the emission probability distribution is represented by GMM (GMM-HMM), using MFCCs as features [50, 51]. In the GMM-HMM approach, each sound event, as well as the periods of silence between events, is modeled by an HMM, and the maximum likelihood path is determined using the Viterbi algorithm. This approach typically achieves only limited performance, however, and it also requires heuristics like the number of simultaneously active events.

Another approach is the use of non-negative matrix factorization (NMF) [52–55], in which a dictionary of basis vectors is learned by decomposing the spectrum of each single-sound event into the product of a basis matrix and an activation matrix, and then combining the basis matrices of all the sound events. The activation matrix for testing is estimated using the combined basis vector dictionary, and is then used either for estimating sound event activations or as a feature vector that is passed on to a classifier. These NMF-based methods can achieve good performance, but they do not take correlations in the time direction into account, instead performing frame-by-frame processing. As a result, the prediction results lack temporal stability so that extensive post-processing is needed. The optimal number of bases for each sound event also must be identified.

More recently, methods based on neural networks have been developed, which have also achieved good SED performance [56–62]. A single network is typically trained to solve a multi-label classification problem involving polyphonic SED. Some studies [57, 59, 60, 62] have also utilized recurrent neural networks (RNN), which consider correlations in the time direction. Although these approaches achieve good performance, they must still perform frame-by-frame detection and they do explicitly model the duration of the output label sequence. Additionally, threshold values for the actual outputs need to be determined carefully to return the best performance.

The SED methods described above are trained using ground-truth data that represents the precise start and end timestamp of each sound event. However, such precise ground truth data is difficult to prepare, even with a human labeling a recording, especially in the case of real world recordings. To address this issue, the use of weakly-labeled data has been explored recently. Weakly-labeled data contains only information about the existence and order of each sound event. In other words, it does not contain detailed time stamp information. Some studies [63, 64] have tackled this challenging problem using convolutional recurrent neural networks (CRNN), however, the performance is still much worse than methods that use ground-truth data with precise time stamp information.

Another problem related to SED is how target events should be defined. Although this definition is highly dependent on the type of application, limiting the range of targets to discrete sound events alone is insufficient for tasks like life-logging, which requires the identification of common human activities, and one activity may generate many different sounds. Thus,

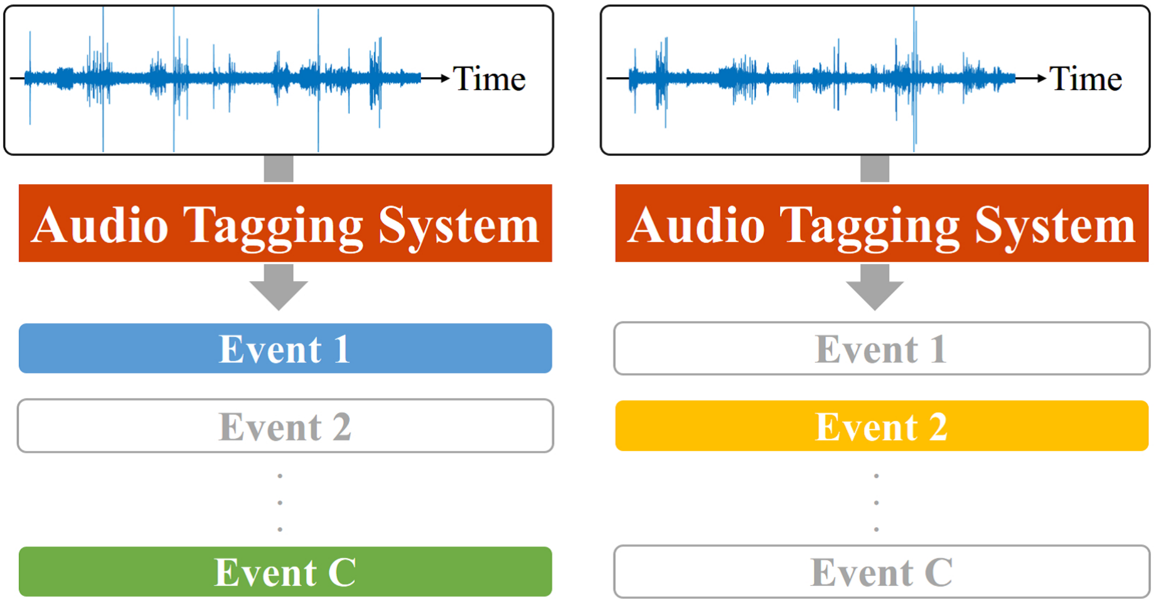


Figure 2.4: *Overview of an audio tagging system.*

another signal should be used as additional information since the activities monitored by a life-logging systems involve sound and also motion. In the field of human activity recognition (HAR), which is related to SED, acceleration signals have been used to recognize human activities like walking, running, and climbing stairs [65–68]. Other studies [69–71] have utilized multi-modal signals recorded with acceleration sensors, gyro sensors, microphones, geomagnetic sensors to recognize more complex activities.

## 2.4 Audio Tagging

Audio tagging is the detection all the predefined sound events included in a short acoustic signal, generally 10 seconds long. In other words, audio tagging is a multi-label classification problem that can be though of a simpler version of polyphonic SED, in that it identifies sound events but does not perform timestamp estimation. An overview of an audio tagging system is shown in Fig. 2.4. Audio tagging can be used for applications such as the indexing of movie clips for retrieval [9] and life-logging [10,11]. Since there is no timestamp information in the ground-truth data, audio tagging systems need to consider the entire audio sequence to detect sound events. Hence, the reference time is relatively long.

One typical audio tagging approach is to train a binary classifier for each sound event. One study created bag-of-words using a sequence of frame-wise features and then trained classifiers, such as SVM and Boosting, for each sound event using the bag-of-words features [72]. Another study trained a pair of GMMs with MFCCs for each sound event, so that one GMM modeled the probability that the event was present in the signal, and the other modeled the probability of the event's absence [73]. A score is then calculated using the log-likelihood ratio for each trained GMM.

Another approach to audio logging is based on the use of a neural network to perform multi-label classification [74, 75]. In these studies, a single neural network with a sigmoid function as the activation function of the output layer is trained to solve the multi-label classification problem. In [74], a CNN with constant-Q transform features was applied for audio tagging. Another study [75] utilized representation learning by employing a de-noising auto encoder (DAE) to extract bottle-neck features.

## 2.5 Anomalous Sound Event Detection

Anomalous SED is the task of detecting the start and the end times of anomalous sound events that do not appear in the training data. An overview of an anomalous SED system is shown in Fig. 2.5. The length of the reference time for anomalous SED is the shortest of the SED tasks, however, the degree of identification is low because the anomalous sound event is generally not predefined. Thus, it is necessary to build the system in an unsupervised manner. Note that anomalous SED is closely related to acoustic novelty detection and acoustic

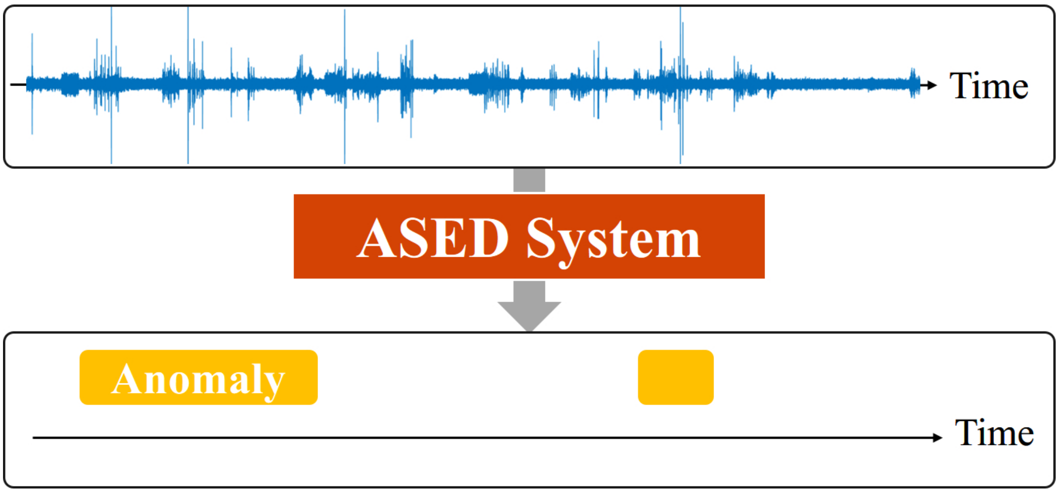


Figure 2.5: *Overview of an anomalous sound event detection system.*

anomaly detection.

One typical approach to anomalous SED is change-point detection [76–78], which involves the comparison of a model of the current time segment with that of a previous time segment and calculating the dissimilarity score. Highly dissimilar comparison results are then used to identify anomalies.

Another approach is outlier detection [79–81], which models an environment's normal sound patterns, and then detects patterns which do not correspond to the normal model, identifying them as anomalies. These normal patterns are those patterns that appear in the training data. Typically, a GMM or a one-class SVM with acoustic features, such as MFCCs, is used [82, 83].

Thanks to recent advances in deep learning, neural network based methods for anomalous SED have been attracting attention [84–86]. An auto-encoder (AE) or long short-term memory recurrent neural network (LSTM-RNN) can be trained using only normal scene data. The AE encodes the inputs as latent features and then decodes them as original inputs, and LSTM-RNN predicts the next input from the previous input sequence. Using a trained model, reconstruction errors between the observations and predictions are calculated, and high-error patterns are identified as anomalies. Although these methods have achieved good performance, it is difficult to model acoustic patterns directly in the time domain due to the highly non-stationary nature and the high sampling rates, so feature vectors extracted from audio signals are typically used instead.

## 2.6 Summary

This chapter provided an overview of the field of environmental sound understanding and its applications. Four major tasks that researchers have focused on in the field of environmental sound understanding were also introduced, and various techniques that have been used to perform each task were reviewed.

Recent advances in machine learning have improved capacity for performing each of these tasks, but the level of performance that will be needed for practical applications remains to be found. There is still much room for improvement and problems encountered when performing each of these tasks will need to be resolved to further enhance performance.

# 3 Human Activity Recognition based on Deep Neural Network Using Multi-modal Signals

This chapter describes human activity recognition based deep neural network using multi-modal signals. Toward the development of a life-logging system, this work focuses on the use of multi-modal signals recorded under realistic conditions so that sound events related to common human activities can be detected, including discrete sound events like door closing along with sounds related to more-extended human activities like cooking. The key to realizing these applications is finding associations between different types of signals that facilitate the detection of various human activities. Below, two deep neural network (DNN)-based fusion methods are proposed to join with multi-modal signals together. Then, a large-scale database of human activities recorded in the real world is created for testing the performance of the proposed methods. This database consists of over 1,400 hours of data including both the outdoor and indoor daily activities of 19 subjects under practical conditions. Furthermore, to address the problem of model individuality, speaker adaptation techniques used in the field of automatic speech recognition are introduced. Experimental evaluation using the constructed database demonstrate that the use of multi-modal signals is effective, and that speaker adaptation techniques can improve performance, especially when using only a limited amount of training data.

## 3.1 Introduction

The goal of human activity recognition (HAR) system is to identify human activities from observed signals. These systems have great potential in various applications such as life logging [87], monitoring the elderly [88], detection of wandering behavior in dementia patients [89], health care [10, 90], and control systems in automated "smart homes" (light switches, climate control, etc.) [12, 91] The mass marketing of electronic devices with sensing capabilities has made it possible to easily acquire various types of signals which can be used to identify the human activity, and as a result, the field of HAR has been attracting more attention.

HAR can be divided into two main categories: environmental augmentation approach and wearable sensing approach. The first approach, environmental augmentation, utilizes information collected with sensors embedded in an environment to recognize subjects' activities. In the field of computer vision, cameras have been utilized to detect subjects' physical activities [92, 93], or to understand group activities [94, 95]. On the other hand, in the field of environmental sound understanding, microphones have been utilized to identify sound events such as phone ringing, typing on a keyboard, and human speech [51, 55]. These approaches allow recognition of various types of activities, however, there is a limitation of installation location. Furthermore, the use of cameras may subjects uncomfortable due to lack of privacy. Another approach of environmental augmentation is based on the use of ubiquitous sensors such as radio frequency identifier (RFID) tags and switch sensors [96–99]. In these approaches, with the embedding small sensors to all of the objects in a room, the system can not only detect the use of objects such as a knife, spoon, and cups but also recognize complicated human activities such as making coffee, taking a medicine and washing dishes. However, they require the embedding of many sensors, making it very costly.

The second approach, wearable sensing, utilizes information collected with wearable sensors attached to a subject's body to recognize activities, especially which have characteristic motion or sound. Compared to the environmental augmentation approach, wearable sensor approaches generally involve much lower costs because it does not require the embedding of many sensors. One of the most typical approaches is based on the acceleration signals recorded with wearable devices to recognize the activities such as walking, running, cycling, and going up (or down) the stairs [65–68]. However, it is difficult to recognize the complicated activities including the use of objects. To address this issue, some studies have combined various type of sensors such as an acceleration sensor, gyro sensor, microphone, geomagnetic sensor [69–71].

Recently, with the improvements in deep learning and the advent of public benchmarking datasets [100, 101], neural network based approaches to HAR in wearable sensing have been proposed. In the study [102], feed-forward neural networks with human-designed features have been utilized, outperforming conventional classification methods such as support vector machine (SVM). Other studies have utilized deep convolutional networks to extract features from observed signals, and long short-term memory (LSTM) recurrent neural networks to capture the temporal dependencies between activities, achieving great performance without the use of human-designed features [103, 104]. However, these approaches were evaluated using a huge database which was recorded in a sensory-rich environment, with subjects who attached many sensors to their bodies. The performance under the practical situation, where only limited sensors or a limited amount of data are available, has not been evaluated. Moreover, a realistic HAR system must be able to handle unknown subjects with only a small amount of subject-specific data. Therefore, it is also necessary to evaluate the performance under these conditions.

In this work, toward the development of the monitoring system for life-logging (Fig. 3.1), two deep neural network (DNN)-based fusion methods with multi-modal signals are pro-

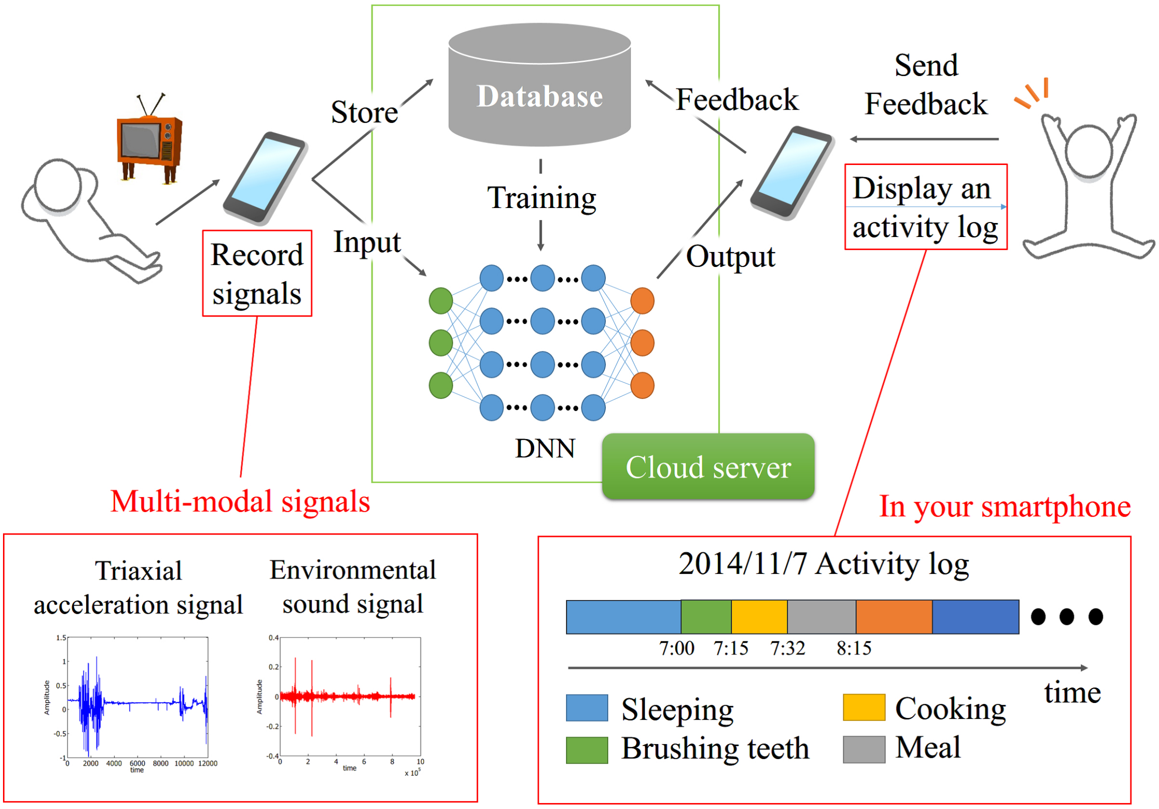


Figure 3.1: *Overview of the target life-logging system. The system uses a smartphone to continuously record environmental sound and acceleration signals, and sends these signals to a server. In the server, the subject's current activity is automatically recognized by an activity recognition model, and then recognition results are then sent to the subject's smartphone. The subjects can not only view their activity history but also send feedback to improve the recognition performance.*

posed and then a large-scale human activity database is created for testing the performance of the proposed methods. The database consists of over 1,400 hours of data including both the outdoor and indoor daily activities of 19 subjects under practical conditions. Furthermore, the problem of model individuality is addressed, which results in degradation of performance when a model constructed for a particular subject is used to attempt to classify the activities of another individual. To do this, speaker adaptation techniques used in the field of automatic speech recognition are introduced. Experimental evaluation using the constructed database demonstrate that the use of multi-modal signals is effective, and that speaker adaptation techniques can improve performance, especially when using only a limited amount of training data.

The rest of this chapter is organized as follows: Section 3.2 describes the construction of the human activity database. Section 3.3 describes the proposed fusion methods and their adaptation methods. Section 3.4 describes the design of the experiment and evaluate the performance of the proposed methods. Finally, this chapter is summarized in Section 3.5.

## 3.2 Nagoya-COI Daily Activity Database

This section describes the construction of Nagoya center of innovation (Nagoya-COI) daily activity database.

### 3.2.1 Recording condition

The outline of recording condition of daily activity is shown in Table 3.1, and the equipment of subject is shown in Fig. 3.2. An accelerated signal is recorded with a smartphone put in a pocket of the rear of the subject's trousers, and an environmental sound signal and a video are recorded with a small video camera attached to the subject's shoulder. The

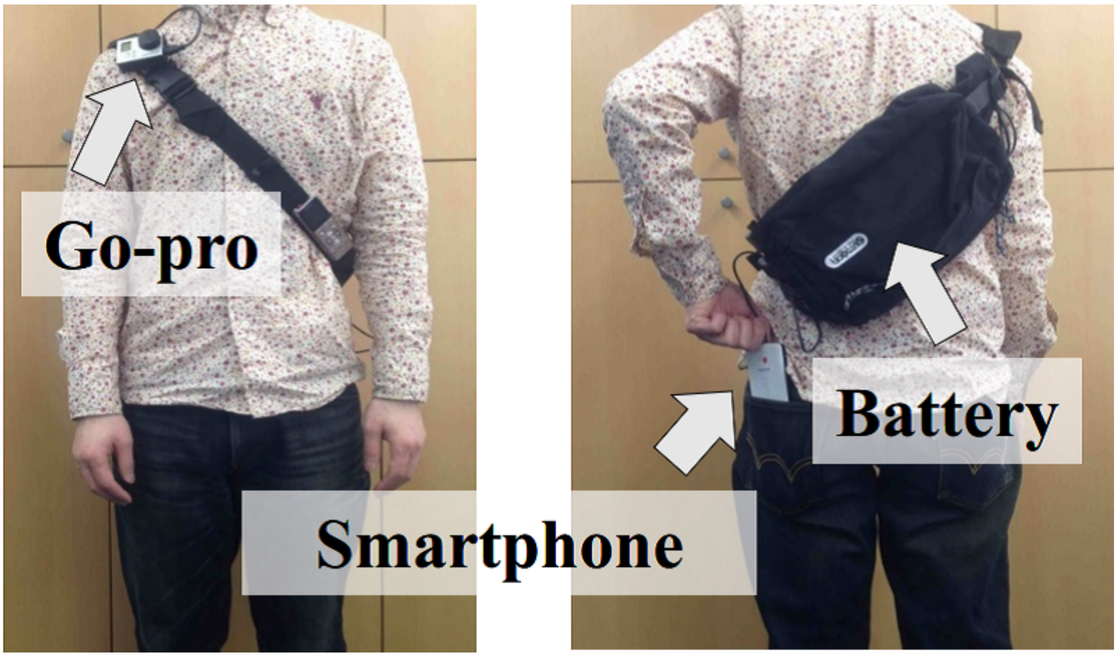


Figure 3.2: *Recording equipment worn by subjects. Note that the video camera is only for data annotation purposes and is not part of the target system.*

Table 3.1: *Data recording conditions.*

| | |
|---|---|
| Number of subject | 1 (long-term) + 18 (short-term) |
| Recoding environment | one-room studio apartment |
| Instruction | Lead well-regulated life |
| Recorded signals | Triaxial acceleration signals (200 Hz) |
| | Environmental sound signal (16k Hz) |
| | Video (1280×720, 29.97 fps) |

recording environment is a one-room studio apartment. Note that it is an apartment with a kitchen/dining area and a separate bedroom. Subjects can freely live in the room, however, they were asked to lead a well-regularized life so that a variety of activity data from each subject could be obtained. In other words, subjects were encouraged to avoid sleeping all day, watching excessive amounts of television, etc. And in order to prevent recording errors, subjects were asked not to go outside alone, but to let an assistant accompany them to help record their outdoor activities.

### 3.2.2 Recorded data

1,400 hours of data including both indoor and outdoor activities are recorded. Then, 300 hours of indoor activity data are annotated, and two types of datasets are constructed: 1) long-term, single subject data of 48 hours in length, 2) short-term, multiple subject data with a total length of 250 hours. The sampling rates of the recorded acceleration signals and environmental sound signals were 200 Hz and 16,000 Hz, respectively. The frame rate of the recorded video was 29.97 fps and resolution was $1,280 \times 720$. The video and environmental sound signals were synchronized, but the acceleration signals were not synchronized because a different recording device was used. Therefore, these signals are synchronized using recording time information from the video and the time stamp information of the acceleration signal. Note that the time stamp information was recorded every sampling, and therefore, it has enough time resolution to synchronize.

### 3.2.3 Annotation

Three people independently annotated the recorded signals using the recorded video and the ELAN annotation tool [105]. After that, another person checked the annotation. Activity

Table 3.2: *Recorded daily activities.*

| Activity name | Length [min] | Activity name | Length [min] |
|---|---|---|---|
| Others | 3,879 | Cleaning | 188 |
| Sleeping | 2,731 | Writing | 150 |
| Note-PC | 2,252 | Cleaning bath | 107 |
| Smartphone | 1,959 | Calling | 104 |
| Watching TV | 1,873 | Tablet | 86 |
| Cooking | 1,827 | Light meal | 85 |
| Eating | 908 | Drying clothes | 75 |
| Clearing table | 679 | Washing | 36 |
| Reading | 476 | Waking | 30 |
| Toilet | 310 | Monologue | 5 |
| Tooth brushing | 214 | Taking a bath | 958 |

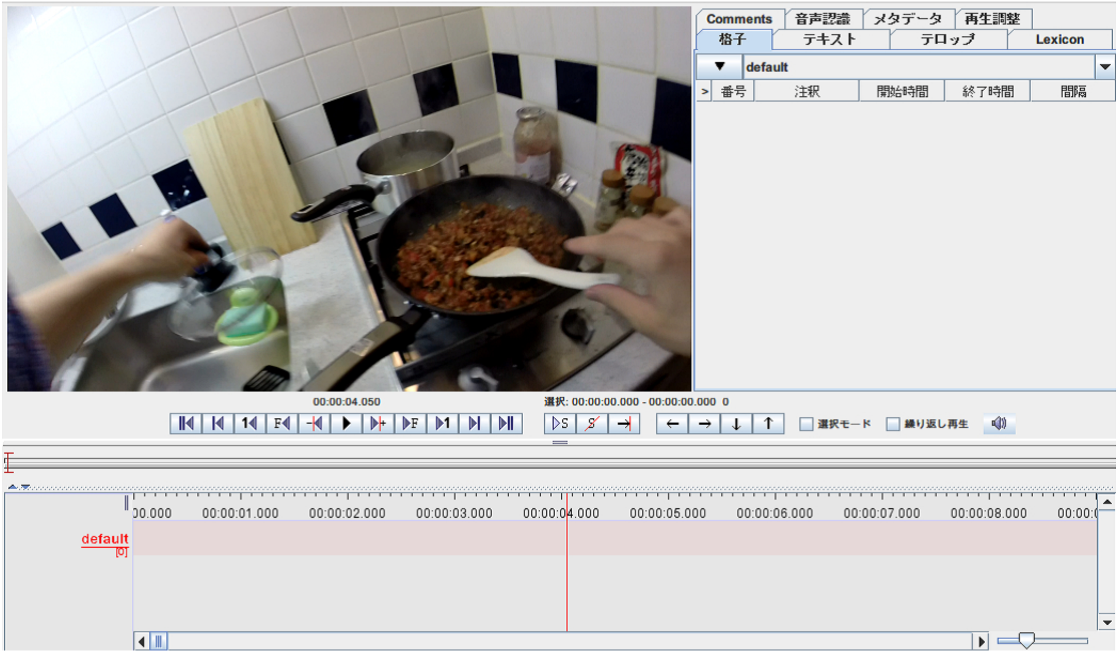


Figure 3.3: *Annotation with ELAN.*

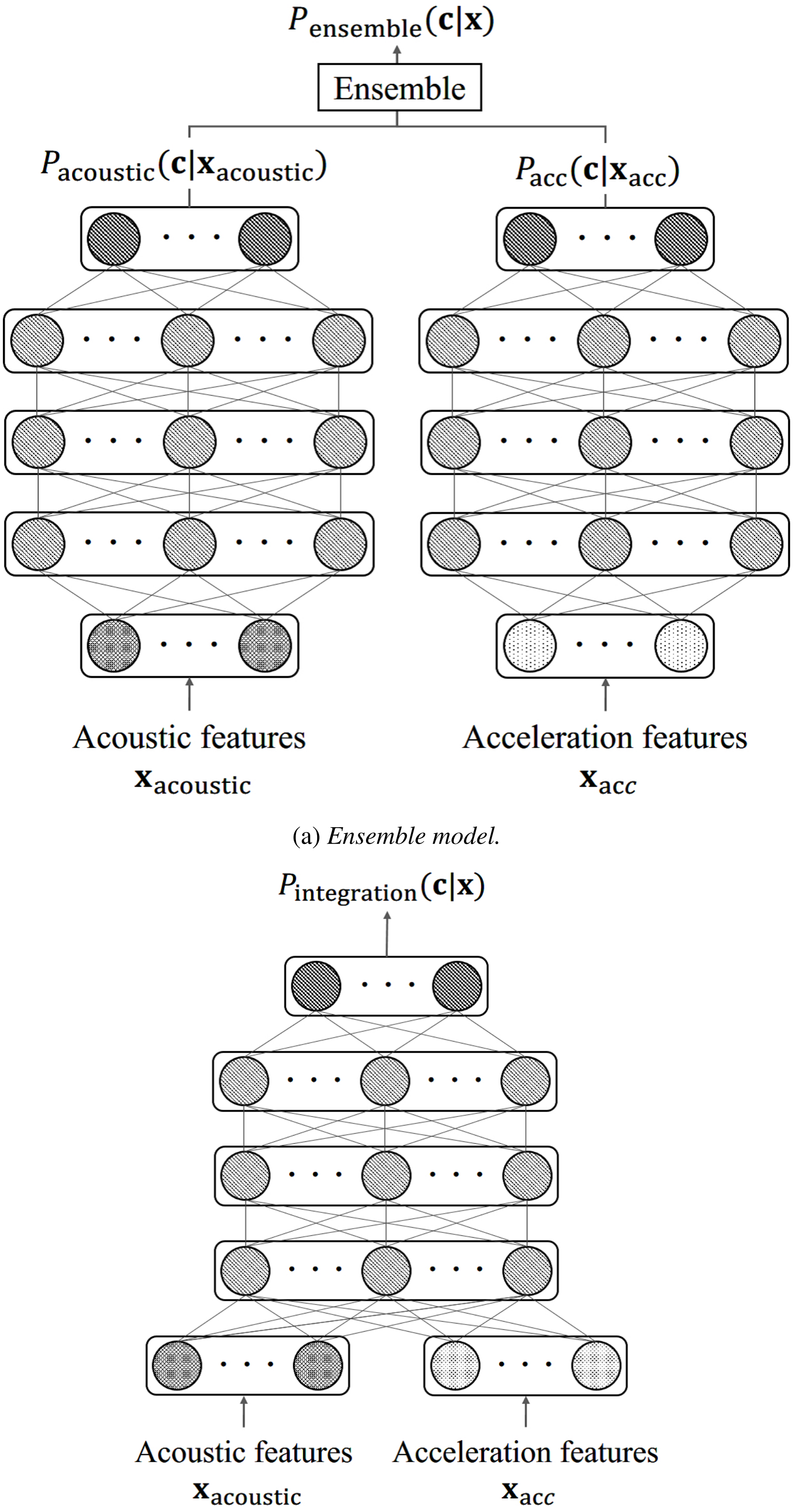
tags used in the annotation and total duration lengths of the individual tag are shown in Table 2. Total 21 tags are used to represent daily activities, and an “Other” tag is used to represent when a subject’s activity could not be determined from the video. All of the activities of the subjects are tagged, which could be determined from the recorded video. When recording the activity of “Sleeping”, the subject wore only the smartphone and the camera placed on the bedside desk. Similarly, when recording the activity of “Taking a bath”, the subject removed all equipment but the equipment was placed in the bathroom and continued recording. There were also situations when subjects conducted multiple activities simultaneously (e.g., eating lunch while watching TV). In these situations, two types of annotations are used: a primary tag to represent the main activity and a secondary tag to represent a sub-activity. In this study, it is assumed that the activity started first was the primary, and that simultaneous activities initiated later were secondary. Finally, to simplify the evaluation experiment, the signals are divided according to their tags, and then they are cut into samples of one minute in length.

## 3.3 Daily Activity Recognition Model

In this section, deep neural network (DNN)-based daily activity recognition model and its adaptation methods are described.

### 3.3.1 Pre-processing and feature extraction

The acceleration signals recorded using a smartphone included pulsive noise signal which were not related to actual movement, and sometimes the signals lacked consecutive samples, which were likely caused by inadequate smartphone processor performance. Because these factors had a negative influence on the analysis of the data, a median filter is applied to remove pulsive noise signal and spline interpolation are performed to project signals which were missing during sampling as pre-processing procedures. After pre-processing, the environmental sound signal and the acceleration signal are divided into synchronous frames of equal duration, and extracted the features from each frame. Frame size and shift size were both 1 second. Three features are extracted from each environmental sound signal frame: 1) Mel Frequency Cepstral Coefficients (MFCC) + Power + $\Delta$ + $\Delta\Delta$, 2) Root Mean Square (RMS) and 3) Zero-Crossing Rate (ZCR). Finally, 41-dimensional acoustic features for each frame are obtained. MFCC is a feature which reflects human aural characteristics and is often used for speech recognition, and its effectiveness has also been confirmed in acoustic event detection [51]. RMS and ZCR represent volume and pitch, respectively. Then, the following five features are extracted from each acceleration signal using the X, Y, and Z axes of each frame: mean, variance, energy, entropy in the frequency domain, and correlation coefficients, where mean and variance are defined as the mean and variance of the raw accel-


$P_{\text{ensemble}}(\mathbf{c}|\mathbf{x})$
Ensemble
$P_{\text{acoustic}}(\mathbf{c}|\mathbf{x}_{\text{acoustic}})$
$P_{\text{acc}}(\mathbf{c}|\mathbf{x}_{\text{acc}})$
Acoustic features
$\mathbf{x}_{\text{acoustic}}$
Acceleration features
$\mathbf{x}_{\text{acc}}$
$P_{\text{integration}}(\mathbf{c}|\mathbf{x})$
Acoustic features
$\mathbf{x}_{\text{acoustic}}$
Acceleration features
$\mathbf{x}_{\text{acc}}$

(a) *Ensemble model.*

(b) *Integrated model.*

Figure 3.4: *Proposed DNN activity classifier.*

eration signal. The mean represents the orientation of the smartphone, and is closely related to the user's posture. For example, suppose that the smartphone is put in a rear pocket of the user's trousers. When the user is standing, the Y axis acceleration component includes acceleration forces due to the earth's gravity. However, when the user is sitting, the Y axis acceleration component omits the effect of gravity. The variance represents the intensity of a user movement, and is effective for detecting user movement. Energy E represents the sum of the absolute values of the fast Fourier transform (FFT) components excluding the DC component, as expressed by the following equation:

$$E = \sum_{i=1}^{N-1} |F_i|^2, \tag{3.1}$$

where $F_i$ represents the $i$-th FFT component of the signal of each axis. This feature is also effective for detecting user movement. Entropy in the frequency domain is represented as follows:

$$S = -\sum_{i=1}^{N-1} p(i) \log p(i), \tag{3.2}$$

where $p(i)$ represents the probability distribution derived from the normalized FFT component using the following equation:

$$p(i) = \frac{|F_i|^2}{\sum_{j=1}^{N-1} |F_j|^2}. \tag{3.3}$$

This entropy value enables us to discriminate between different activities which have the same intensity. Correlation coefficient $r$ between two axis is defined for the series data $\mathbf{s}_1, \mathbf{s}_2$ of two axis as follows:

$$r(\mathbf{s}_1, \mathbf{s}_2) = \frac{\mathrm{Cov}(\mathbf{s}_1, \mathbf{s}_2)}{\sigma_{\mathbf{s}_1} \cdot \sigma_{\mathbf{s}_2}}, \tag{3.4}$$

where $\mathrm{Cov}(\mathbf{s}_1, \mathbf{s}_2)$ represents covariance between two vectors and $\sigma$ represents a standard deviation of vector components. The correlation coefficients represent the direction of movement of the smartphone, which is related to user movement.

Finally, all of these features extracted from the sound and acceleration signals are concatenated and a total of 56 dimensional features are used as classifier inputs.

### 3.3.2 Activity classifier

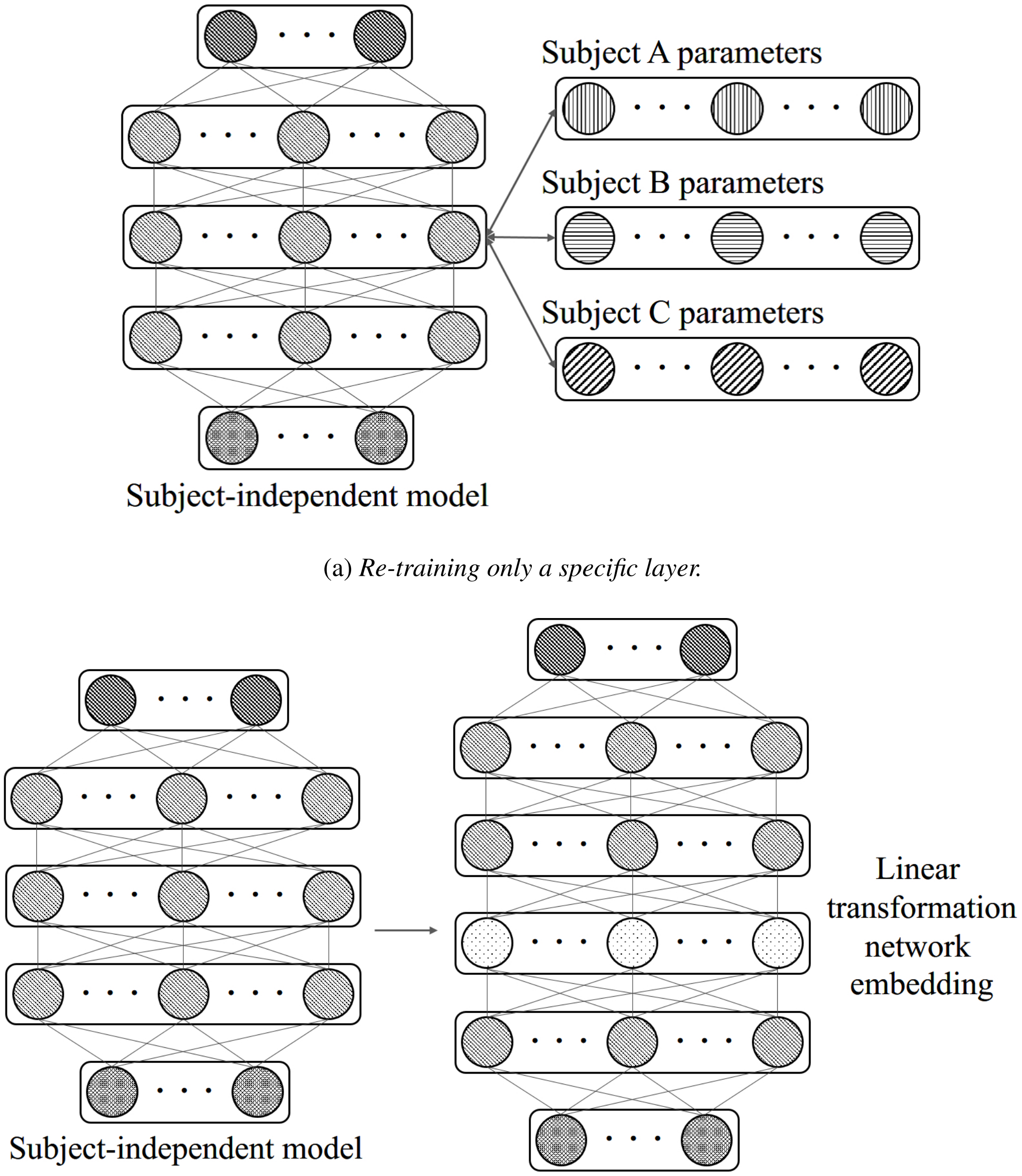
In this study, DNNs are used as an activity classier. DNNs can not only deal with high dimensional feature vectors reflecting a time sequence, but can also be trained to automatically convert themselves into discriminative feature vectors through lamination of their hidden layers.

In this study, two types of DNNs are introduced: 1) an ensemble model which integrates the outputs of the acoustic and acceleration feature models, and 2) an integrated model which utilizes acoustic and acceleration features as an input feature vector. The structures of these DNNs are shown in Fig. 3.4. The outputs of ensemble model $P_{\text{ensemble}}(c|\mathbf{x})$ are calculated as follows:

$$P_{\text{ensemble}}(c \mid \mathbf{x}) = P_1(c \mid \mathbf{x}_{\text{acoustic}})^w P_2(c \mid \mathbf{x}_{\text{acc}})^{1-w}, \tag{3.5}$$

where $P_1(c \mid \mathbf{x}_{\text{acoustic}})$ and $P_2(c \mid \mathbf{x}_{\text{acc}})$ are outputs of the acoustic feature model and the acceleration feature model, respectively, $c$ is an index of activity class, $w$ is a weight coefficient between the acoustic feature model and the acceleration feature model, and $\mathbf{x}_{\text{acoustic}}$, $\mathbf{x}_{\text{acc}}$, and $\mathbf{x}$ are the acoustic feature vector, the acceleration feature vector, and the concatenated feature vector, respectively. The networks consist of 3 hidden layers with 2,048 hidden nodes, and a sigmoid function is used as an activation function. The number of nodes in the input layer corresponds to the dimensions of the input feature vector, and the number of nodes of the output layer corresponds to the number of target activity classes.

The training procedure is as follows. First, the features of 11 frames are concatenated, which included a center frame, the 5 preceding frames and the 5 succeeding frames, utilizing


Subject A parameters
Subject B parameters
Subject C parameters
Subject-independent model
Subject-independent model
Linear transformation network embedding

(a) *Re-training only a specific layer.*

(b) *Linear transformation network embedding.*

Figure 3.5: *Outline of adaptation methods*

a key property of DNNs which is the ability to deal with large numbers of dimensional feature vectors. Second, the concatenated features are normalized using all of the training data, making the mean and the variance of each dimension 0 and 1, respectively. Third, the DNN is pre-trained using greedy learning with a denoising auto encoder (DAE), in order to appropriately set the initial parameters of the DNN using the normalized, concatenated features. When training the DAE, Gaussian noise with a variance of 0.1 is added to the input vectors. Finally, the DNN is fine-tuned with back-propagation using annotation data. During the fine-tuning phase, Adam optimization method [106] and dropout [107] with a fixed learning rate of $5e-4$ are used.

### 3.3.3 Adaptation methods

To achieve better activity recognition performance, it is necessary to build a customized model for each user. However, it is difficult to collect sufficient data for each user, and building the model requires annotations of all the data. Therefore, an adaptation technique used in the field of speech recognition is introduced, which enables to fit a trained model for a specific user even if only a small amount of subject-specific training data is available.

Three types of adaptation methods are used, whose effectiveness has been confirmed in the field of speech recognition [108,109]. The first adaptation method is to train all of the layers of the DNN. When using this approach, the parameters of the subject independent model are used as the initial parameters, and then the network is re-trained using subject-specific data for adaptation. If the amount of subject-specific data used for adaptation is small, the network will tend to over-fit, therefore it is necessary to determine a suitable regularization coefficient. In this study, the coefficient was determined through preliminary experiments.

The second adaptation method is to re-train only a specific layer with subject-specific data, which is selected as a subject adaptation layer [108]. A diagram of this method is shown in

Fig. 3.5 (a). The adaptation is performed as follows:

$$\mathop{\arg\min}_{(\hat{\mathbf{W}}^{(l)},\hat{\mathbf{b}}^{(l)})} E(\mathbf{\Lambda}, \hat{\mathbf{W}}^{(l)}, \hat{\mathbf{b}}^{(l)}) + \frac{\beta}{2}\left(||\hat{\mathbf{W}}^{(l)} - \mathbf{W}^{(l)}||^2 + ||\hat{\mathbf{b}}^{(l)} - \mathbf{b}^{(l)}||^2\right), \tag{3.6}$$

where $l$ is the index of the adaptation layer, $\mathbf{W}$ and $\mathbf{b}$ represent the weight and bias parameters before re-training, respectively, $\hat{\mathbf{W}}$ and $\hat{\mathbf{b}}$ represent the weight and bias parameters after re-training, respectively, $\mathbf{\Lambda}$ represents all of the network parameters, and $\beta$ is a regularization coefficient. The first term represents the error function of the network, and the second term represents a regularization term which prevents leaving too much in common with the original parameters.

The third adaptation method is embedding the linear transformation network (LTN embedding) [109]. A diagram of this method is shown in Fig. 3.5 (b). A linear transformation layer is inserted before a specific layer and only the linear transformation layer is re-trained. When inserting the linear transformation layer, its weight parameters $\mathbf{A}$ and its bias parameters $\mathbf{a}$ are initialized as an identity matrix and a zero vector, respectively. The optimization is conducted based on the following equation:

$$\mathop{\arg\min}_{(\mathbf{A},\mathbf{a})} E(\mathbf{\Lambda}, \mathbf{A}, \mathbf{a}) + \frac{\beta}{2}\left(||\mathbf{A} - \mathbf{I}||^2 + ||\mathbf{a} - \mathbf{0}||^2\right), \tag{3.7}$$

where $\mathbf{\Lambda}$ represents all of the network parameters, and $\beta$ is a regularization coefficient. The first term represents the error function of the network, and the second term represents a regularization term which prevents leaving too much data from the identity matrix and zero vector. Note that the third adaptation method, LTN embedding, has a strong restriction compared to the second adaptation method.

## 3.4 Experimental Evaluation

Experimental evaluation is conducted using the constructed database in order to check the performance of the proposed activity recognition model.

### 3.4.1 Subject-closed experiment

First, a subject-closed experiment was conducted, in which the same subject's data was used in both the training and test phases. The results will represent the performance under the condition where a large amount of subject-specific data can be prepared in advance. The target activities are shown in Table 3.3, where the numbers in brackets represent the length of the recorded data in minutes. For this experiment, a long-term, single person dataset was used, and the most frequently observed nine activities were used as the target activities, while all of the remaining activities were used as a non-target activity. When multiple activities were occurring simultaneously, the primary tag for that sample was only used. A data segment of 60 seconds in length was regarded as one sample, and data segments of less than 60 seconds in length were not used for the experiment.

The experiment was conducted as follows: 1) randomly select 10 samples from each activity data set as test data; 2) train the network using the remaining data as training data; 3) evaluate performance for the model using the selected test data; 4) repeat steps 1-3 ten times. In order to evaluate several different models fairly, the test data selected to evaluate the first model was also used to evaluate the other models. The average F measure was used as the evaluation criterion, and all of the DNNs were trained using the open source toolkit Torch7 [110] with a single GPU (Nvidia GTX 980).

#### Effectiveness of multi-modal signals

To confirm the effectiveness of using multi-modal signals, the performance of the following four models were compared using nine types of target activities:

1. Acceleration feature model (Only Acceleration)
2. Acoustic feature model (Only Acoustic)

Table 3.3: *Target activities in subject-closed experiment.*

| Tag | Length [min] | Tag | Length [min] |
|---|---|---|---|
| Cleaning | 39 | Sleeping | 1,257 |
| Cooking | 108 | Smartphone | 198 |
| Meal | 120 | Toilet | 61 |
| Note-PC | 141 | Watching-TV | 109 |
| Reading | 164 | Other | 582 |

3. Ensemble model (Ensemble)
4. Integrated model (Integration)

All of these models have the same DNN structure with the exception of the input layer. Weight coefficient $w$ in Eq. 3.5 was set at 0.75, which was determined experimentally in order to maximize performance.

The experimental result is shown in Fig. 3.6. *Frame* represents frame F measure at the level of the frame unit. *Sample* represents sample F measure obtained using the majority of the frame recognition results in each sample. When we compare the performance of the single signal models and the multi-modal signal models, we can see that the multi-modal signal models achieved better performance. Therefore, we can say that the use of multi-modal signals is more effective for activity recognition. Second, when we compare the performance of the ensemble and integrated models, we can see that the integrated model achieved better performance. This is because the DNN could extract discriminative features using both acceleration features and acoustic features as inputs.

Based on these results, the integrated model was used as DNN in subsequent experiments.

### Comparison with conventional models

Next, the integrated DNN model was compared to the following four conventional methods:

1. k-Nearest Neighbor (KNN),
2. Gaussian Mixture Model (GMM),
3. Decision Tree (Tree),
4. Support Vector Machine (SVM),

where $K$ in the KNN method was 5, the number of mixtures in the GMM was 10, the kernel function of the SVM was an RBF kernel, and the type of SVM is one-versus-one. The SVM was trained using libSVM [111]. These hyper-parameters were determined through preliminary experiments, and all of these models were trained using the same feature vector, which consisted of both acceleration and acoustic features. Since it is assumed that the target system also receives signals of an ambiguous activity which is difficult to determine, the performance under the following two conditions was evaluated: 1) the activity "Other" is not added to target activities (w/o other), and 2) the activity "Other" is added to target activities (w/ other).

Experimental results are shown in Fig. 3.7. We can see that the DNN-based integrated model performed significantly better than the conventional methods, especially when the activity "Ohter" is added to target activities. The performance of conventional methods such as decision tree or SVM decreased by about 10 points when the activity "Other" was added. This is because the "Other" data is widely distributed and significantly overlapped with the other data in the feature space, and therefore, it is basically difficult to determine the complicated hyperplane. On the other hand, the proposed DNN-based model maintained its recognition performance when the "Other" was added. This result implies that the DNN can

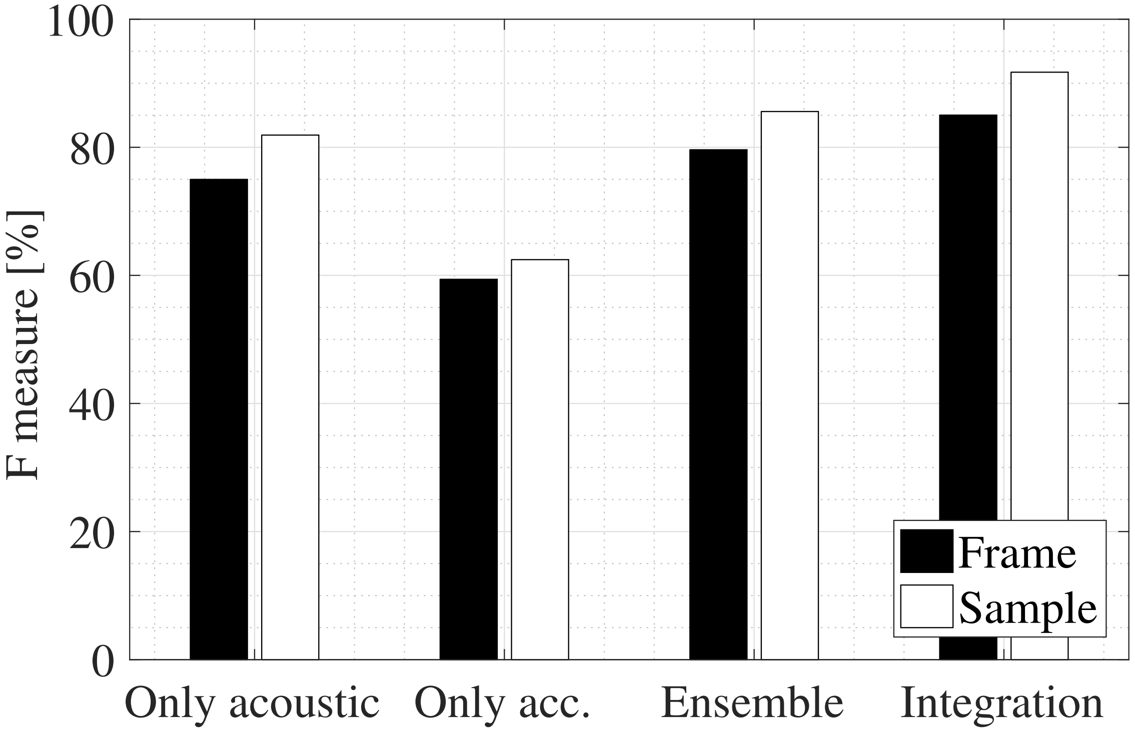


Figure 3.6: *Comparison of four DNN models.*

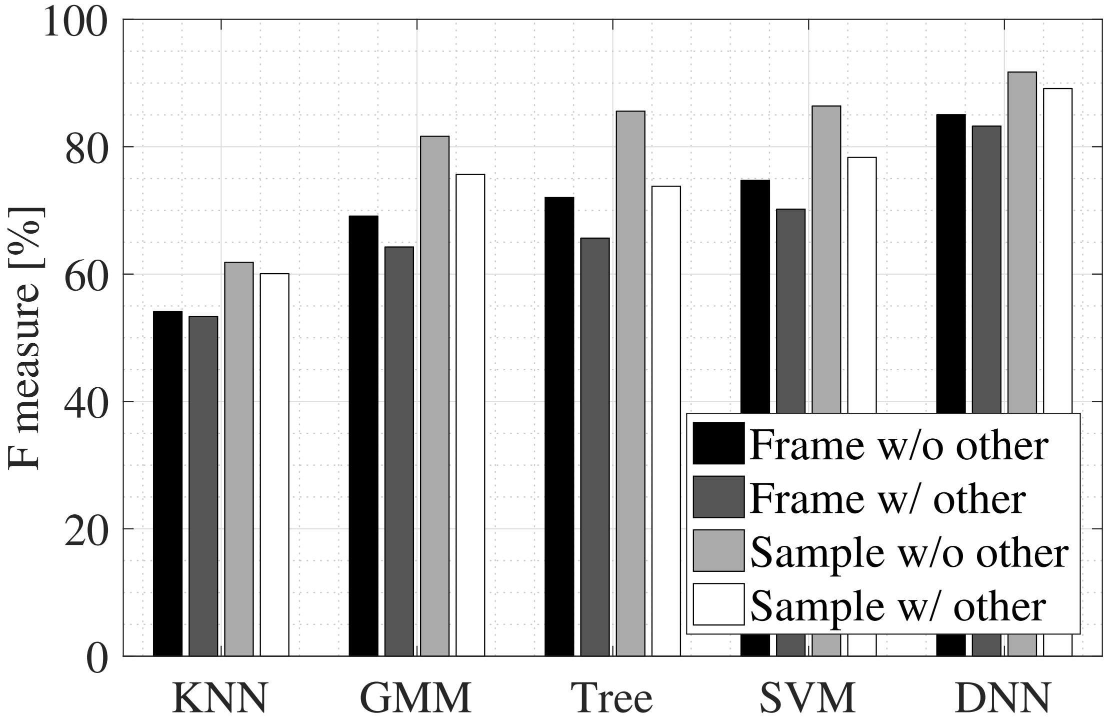


Figure 3.7: *Comparison of proposed DNN model with other conventional methods.*

Table 3.4: *Target activities in subject-open experiment.*

| Tag | Length [min] | Tag | Length [min] |
|---|---|---|---|
| Cleaning | 679 | Sleeping | 2,731 |
| Cooking | 1,826 | Smartphone | 1,959 |
| Meal | 908 | Toilet | 310 |
| Note-PC | 2,252 | Watching-TV | 1,873 |
| Reading | 476 | | |

relatively well model such a complicated hyperplane. A visualization of third hidden layer outputs using t-SNE [112] is shown in Fig. 3.8. We can see that each activity data is distributed separately in the manifold space. This result supports the hypothesis that DNN can model the complicated hyperplane through the conversion to discriminative features using multiple hidden layers.

### 3.4.2 Subject-open experiment

Next, a subject-open experiment was conducted, where the data of different subjects was used in the training and test phases. This experiment evaluates performance when we cannot prepare subject-specific data in advance, and is an important indicator of viability in practical use. For this experiment, a short-term, multiple-subject dataset was used, and the target activities are shown in Table 3.4. The experiment was conducted using leave-one-subject-out validation, where one subject's data is used as test data, and the remaining data of the other subjects is used as training data.

The experimental results are shown in Fig. 3.9 and Table 3.5. From these results, we can see that performance in the subject-open evaluation is much lower than in the subject-

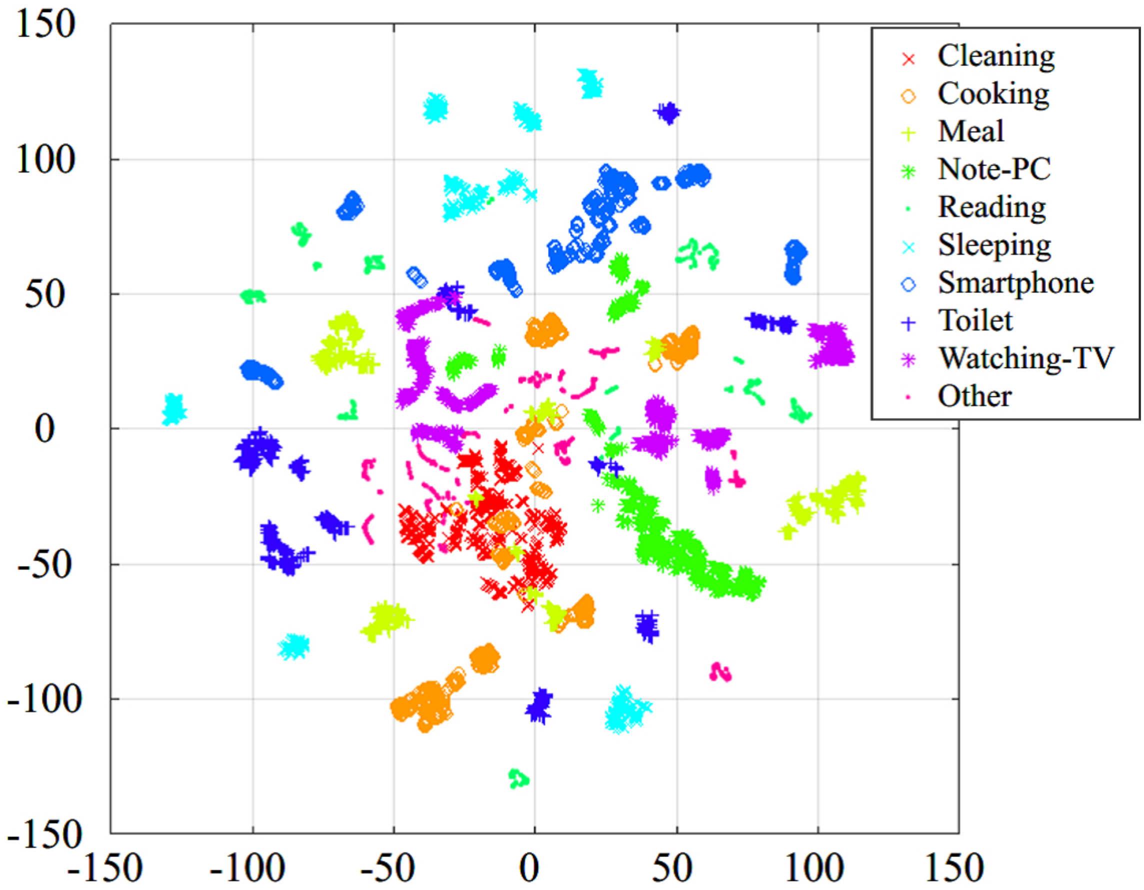


Figure 3.8: *Visualization of third hidden layer outputs using t-SNE.*

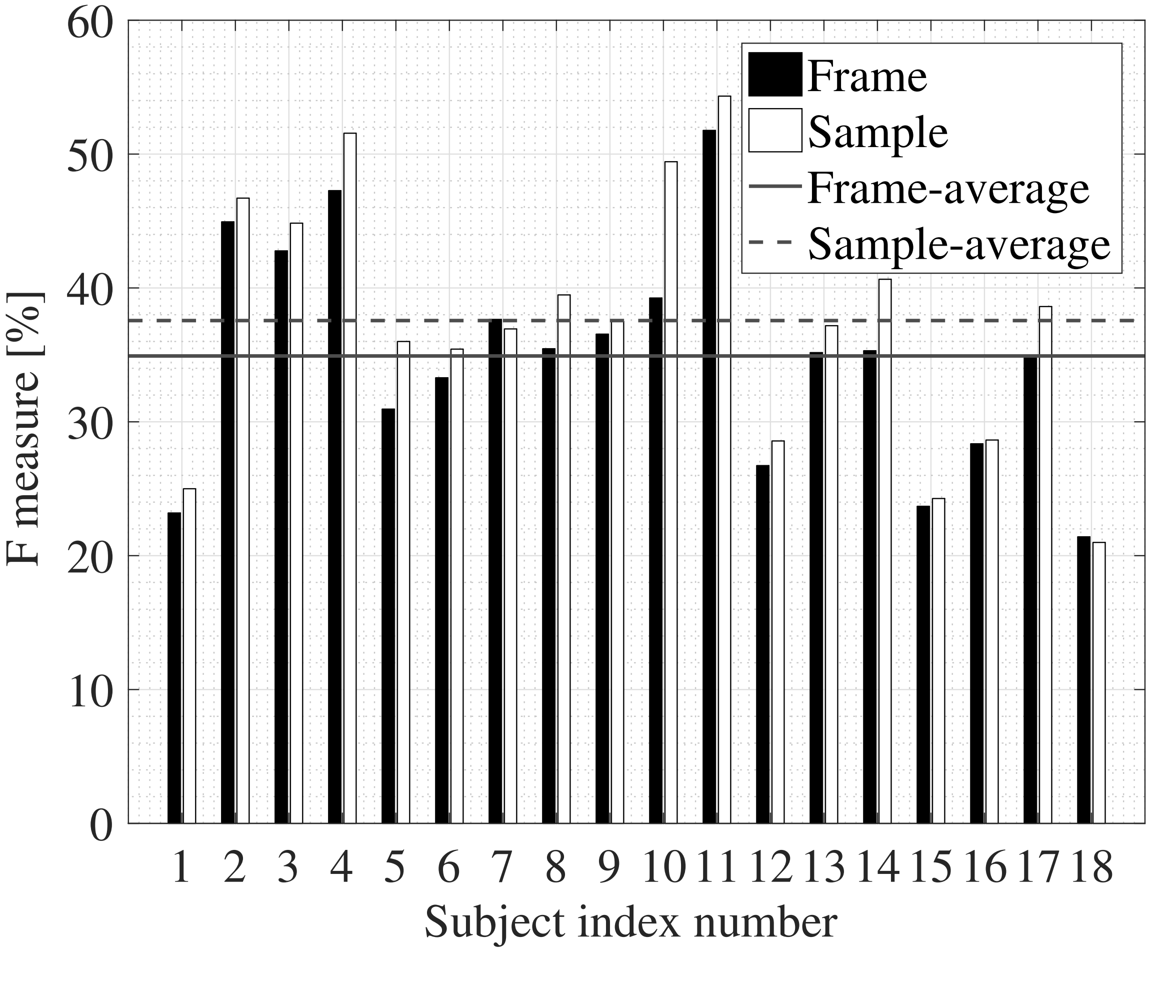
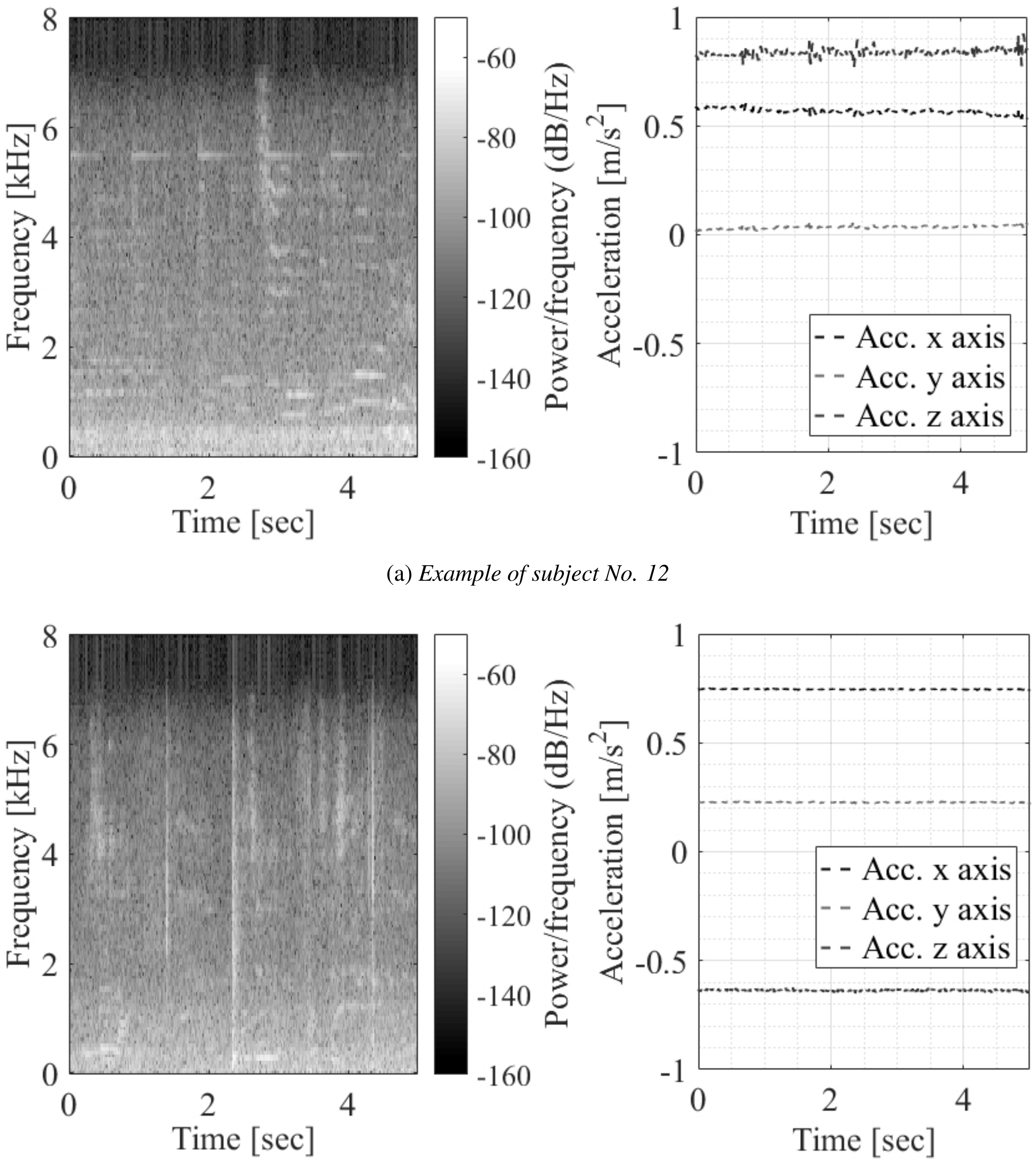


Figure 3.9: *Leave-one-subject-out result*

Table 3.5: *Confusion matrix of subject-open experiment. Diagonal elements represent recall, that of the right end column represent precision, and that of the lower end row represent F measure.*

| Recall | Cleaning | Cooking | Meal | Note-PC | Reading | Sleeping | Smartphone | Toilet | Watching-TV | Precision |
|---|---|---|---|---|---|---|---|---|---|---|
| Cleaning | 69.2 | 28.4 | 0.9 | 0.1 | 0.0 | 0.0 | 0.6 | 0.7 | 0.0 | 41.9 |
| Cooking | 31.3 | 64.0 | 0.5 | 0.5 | 0.2 | 0.8 | 0.2 | 2.2 | 0.3 | 74.6 |
| Meal | 1.0 | 6.1 | 55.9 | 12.6 | 1.2 | 3.1 | 4.2 | 3.2 | 12.8 | 44.3 |
| Note-PC | 0.3 | 1.7 | 10.7 | 22.2 | 15.4 | 16.4 | 5.6 | 3.2 | 24.5 | 40.7 |
| Reading | 0.4 | 1.9 | 9.0 | 13.9 | 6.7 | 19.1 | 8.0 | 21.8 | 19.1 | 3.9 |
| Sleeping | 0.0 | 0.0 | 0.2 | 8.1 | 7.3 | 66.8 | 7.9 | 4.8 | 4.9 | 64.9 |
| Smartphone | 1.0 | 1.5 | 8.2 | 10.5 | 3.9 | 17.8 | 16.7 | 4.7 | 35.7 | 32.6 |
| Toilet | 10.3 | 15.5 | 1.3 | 2.6 | 3.9 | 8.7 | 0.3 | 54.8 | 2.6 | 24.4 |
| Watching-TV | 0.5 | 1.2 | 9.1 | 5.6 | 7.3 | 5.8 | 13.2 | 2.8 | 54.4 | 38.8 |
| F measure | 52.2 | 68.9 | 49.5 | 28.8 | 4.9 | 65.8 | 22.1 | 33.7 | 45.3 | 41.2 |

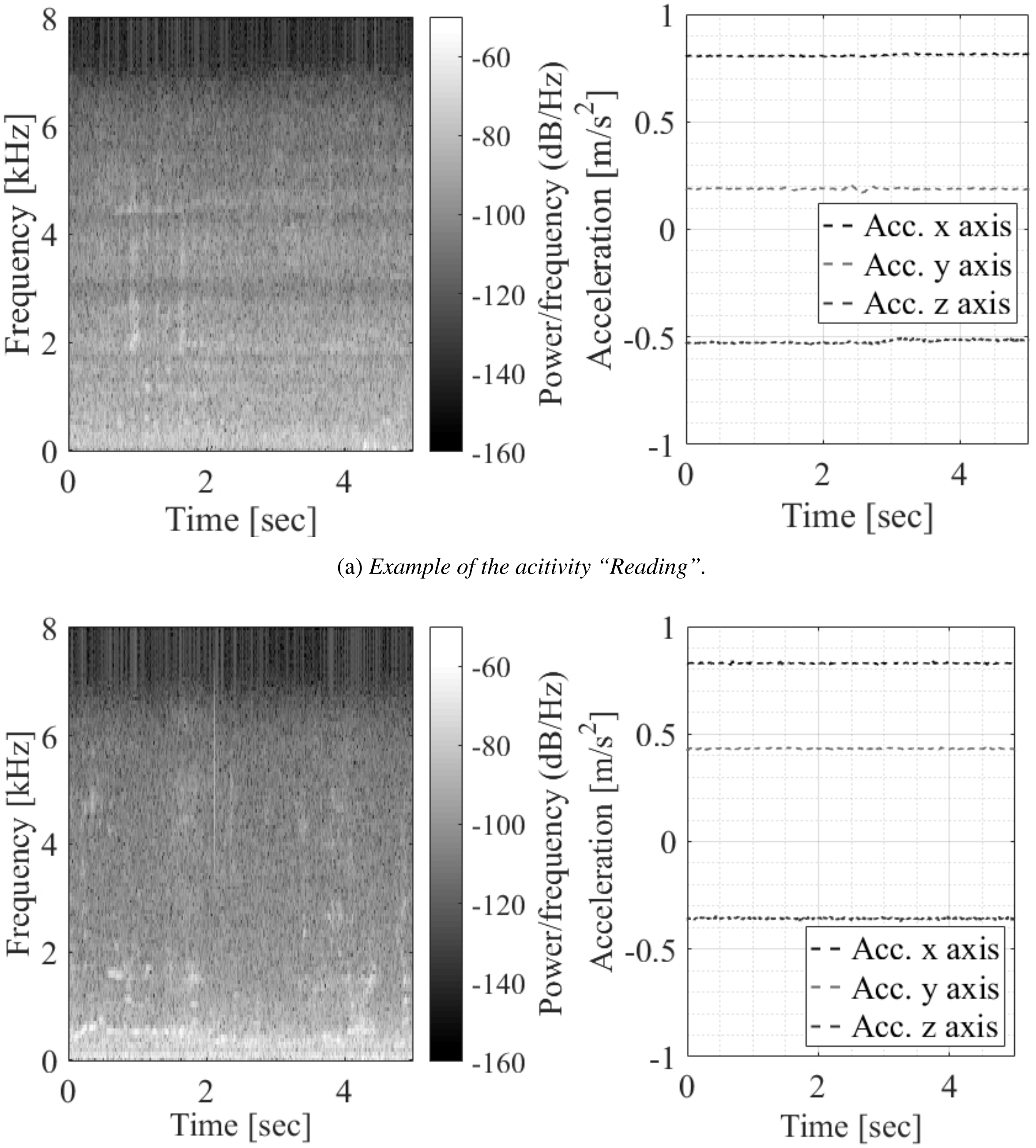
closed experiment, especially for the activities of “Reading”, “Note-PC” and “Smartphone”, for which the model achieved recognition performance of fewer than 20 points. There are two reasons for the poor recognition performance for these activities. First, each subject has a very different way of performing these tasks, i.e., there are large differences in subject behavior, even when they are performing the same activity. Examples are shown in Fig. 3.10, where signals for the activity “Smartphone” are shown for two different subjects. From the figures, we can see that there is a big difference in the recorded signals for each subject as they perform the same activity. Another examples are shown in Fig. 3.11. In this comparison, one subject’s manner of “Reading” is similar to another subject’s manner of “Sleeping”, since the first subject reads while lying on a bed. In addition, such activities do not tend to emit frequent, characteristic sounds, making them harder to detect. A second reason is the lack of uniform orientation of the smartphone in the rear pockets of the subjects’ trousers. Subjects were instructed how to attach the smartphone in their pockets, but some



(a) *Example of subject No. 12*

(b) *Example of subject No. 15*

Figure 3.10: *Example of recorded signals of the activity “Smartphone” for two different subjects. The left figure represents a spectrogram of sound signal, and the right figure represents the recorded acceleration signals.*



(a) *Example of the acitivity "Reading".*

(b) *Example of the activity "Sleeping".*

Figure 3.11: *Example of recorded signals of the activities "Reading" and "Sleeping" for two different subjects. The left figure represents a spectrogram of sound signal, and the right figure represents the recorded acceleration signals.*

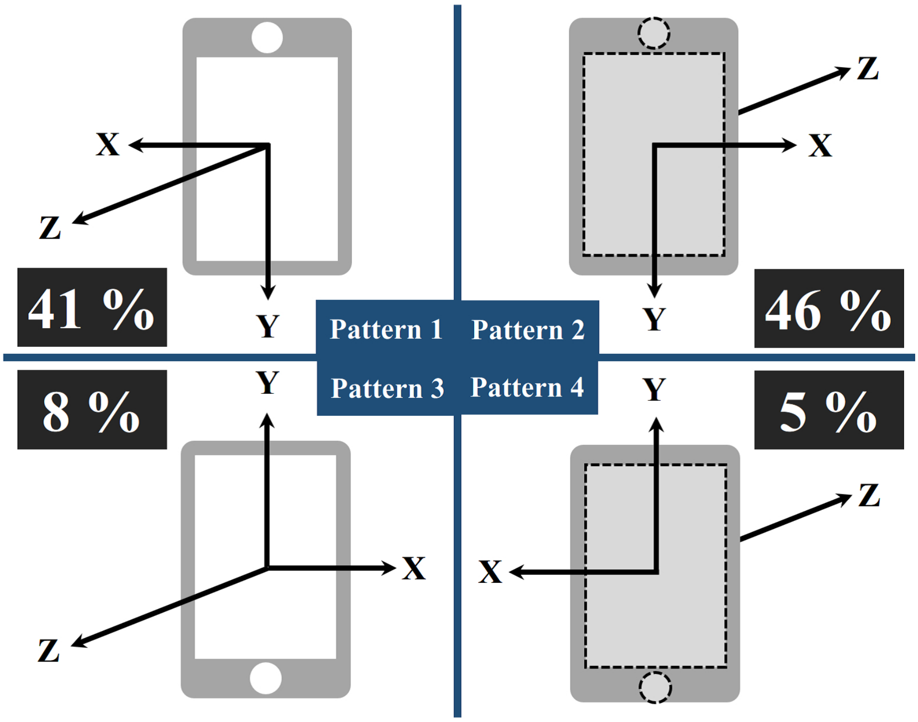


Figure 3.12: *Analysis of smartphone rotation in the rear pocket.*

subjects were not careful about the orientation of the smartphone. To investigate this problem in more detail, data when the subject was standing without making any movements were extracted, and the rotation of smartphone in the rear pocket was divided into four patterns with the mean of acceleration signals. The results are shown in Fig. 3.12, where each axis represents the axis of the noted acceleration signals as recorded by the smartphone in each of the possible orientations, and where the percentages represent the proportion of subjects who positioned their phone in each orientation. From these results, we can confirm that subjects were not careful about the orientation of the smartphone, and that even if subjects perform the same activity in the same manner, recorded acceleration signals will be very different if the smartphone is not in the same position. Therefore, it is necessary to perform a pre-processing to estimate the actual orientation of the smartphone, or extract the new feature which is independent of the orientation of the smartphone.

### 3.4.3 Adaptation experiment

Finally, an adaptation experiment was conducted to see if activity recognition performance could be improved with only a small amount of subject-specific data. To confirm the effectiveness of the adaptations, performance after adaptation was compared with that when the model was constructed using a random initialization. It is expected that performance when using the adaptations will be better than when using a random initialization if the adaptation methods are effective. The adaptation experiment was conducted as follows: 1) build the subject-independent model using a short-term, multiple-subject dataset as training data; 2) randomly select adaptation samples for each activity class from a long-term, single-subject dataset; 3) apply an adaptation method to the subject-independent model using selected samples; 4) evaluate performance using test data selected in the subject-closed experiment; 5) repeat steps 2-4 while increasing the number of adaptation samples. In this experiment, $N = \{1, 2, 3, ..., 25\}$ samples from each activity class for adaptation and 10 samples from each activity class for test data were selected, and these steps were repeated ten times. When the second adaptation method, re-training only a specific layer, was applied, the second hidden layer was selected as the adaptation layer, and the regularization coefficient in Eq. 3.6 was set to $1e - 6$. When the third adaptation method, LTN embedding, was applied, a linear transformation layer was inserted before the second hidden layer, and the regularization coefficient in Eq. 3.7 was set to $5e - 6$. These adaptation layers and regularization coefficients were determined through preliminary experiments.

Experimental results are shown in Fig. 3.13. We can see that performance when using the adaptation methods is better than when using random initialization. This result shows that all of the adaptation methods are effective even if only a small amount of subject-specific

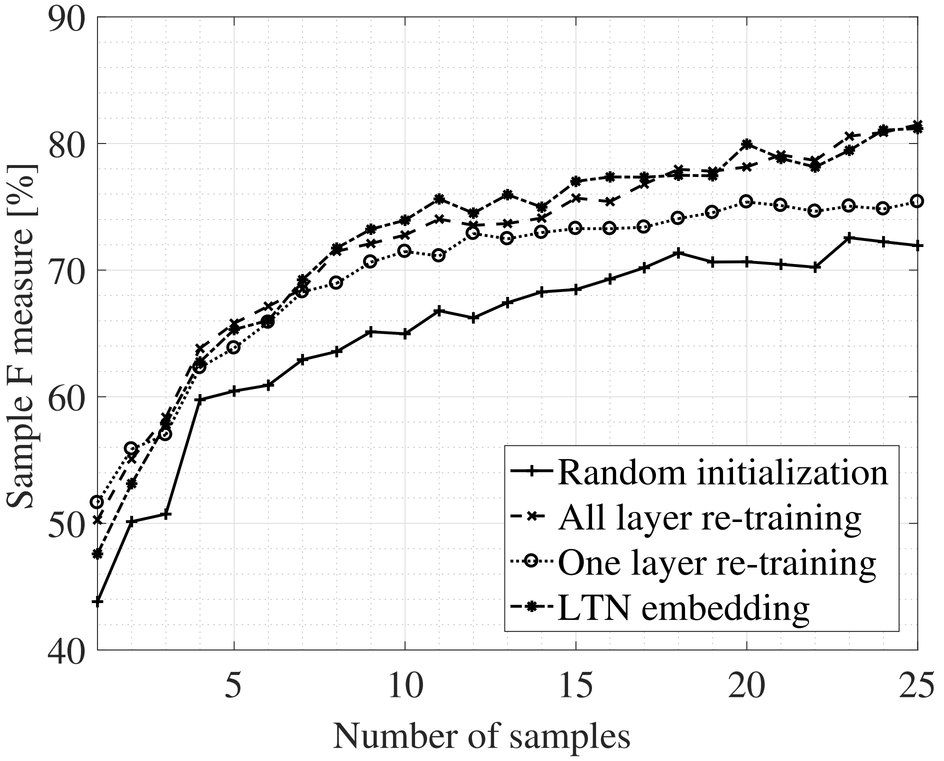
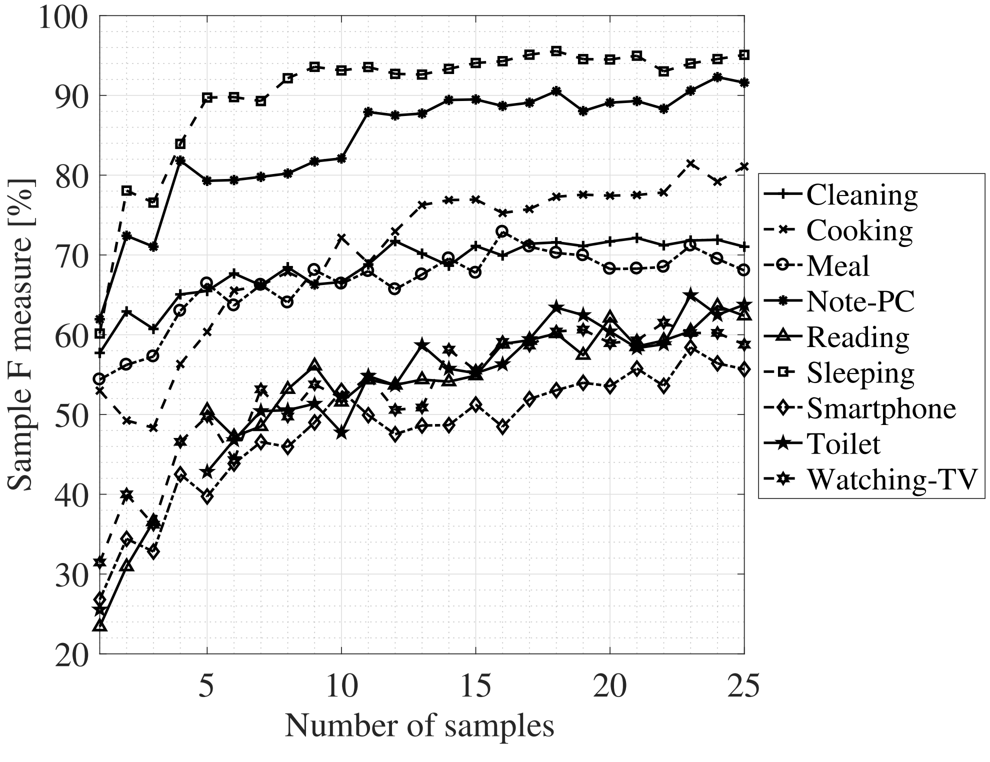


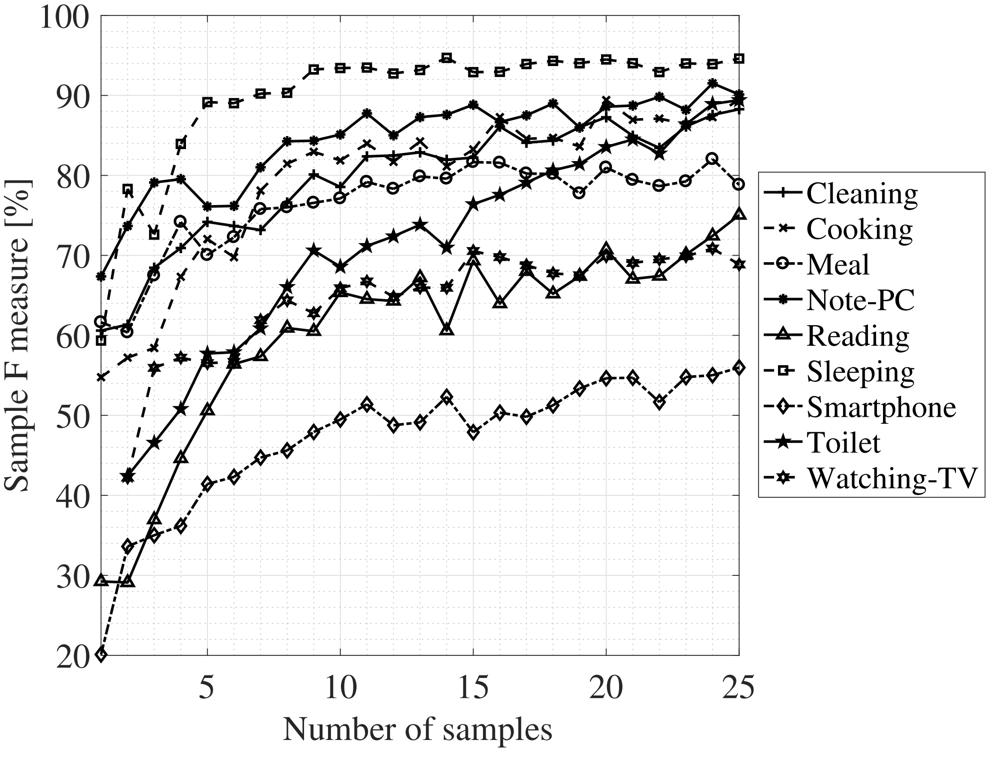
Figure 3.13: *Class average performance using various adaptation methods and different numbers of samples.*

training data is available. In Fig. 3.13, when the number of adaptation samples is from one to ten, there is no difference in performance between these adaptation methods. However, when the number of adaptation samples is more than ten, performance with re-training only a specific layer tends to become saturated. Hence, it is better to use other adaptation methods if we can prepare adaptation data for more than ten. There is no significant difference between performance using all layer re-training and when using LTN embedding, however, the number of parameters to keep for each subject are significantly fewer when using LTN embedding than when using all layer re-training. Therefore, we can say that LTN embedding is a suitable adaptation method in terms of not only improving performance, but also requiring limited computational resources.

Finally, the change in classwise performance is shown in Fig. 3.14, where the graph on the left represents the change in the activity recognition rate when using random initialization, and the graph on the right represents that when using LTN embedding as an adaptation method. By comparing these results, we can see that the adaptation method is effective for improving the accuracy of most of the activities, but not for activities such as “Sleeping”, “Note-PC”, and “Smartphone”. For “Sleeping” and “Note-PC”, even if we use a random initialization the performance is nearly 90 points, therefore there is little room for improvement using an adaptation method. Recognition performance for “Smartphone” was poor even when an adaptation method was used. One reason for this poor performance is that there was a great deal of inconsistency in the manner in which subjects used their smartphones (See the example in Fig.3.10). Therefore, using other subject’s data have no advantage to recognize the activity “Smartphone”.



(a) Random initialization



(b) LTN embedding

Figure 3.14: *Change in performance for various activities when using different number of samples w/o and w/ adaptation.*

## 3.5 Summary

In this chapter, toward the development of smartphone-based monitoring system for life logging, two DNN-based fusion methods with multi-modal signals were proposed and then a large-scale human activity database was created for testing. The database included over 1,400 hours of data including both outdoor and indoor daily activities of 19 subjects in practical situations. Furthermore, speaker adaptation techniques in the field of speech recognition were introduced to address the problem of model individuality. Experimental evaluation using the constructed database demonstrated that the use of multi-modal including an environmental sound and acceleration signals with DNN is effective for the improvement of performance, and that DNN can discriminate specified activities from a mixture of unspecified activities. Furthermore, DNN-based adaptation methods could improve performance, especially when only a limited amount of subject-specific training data is used.

# 4 Polyphonic Sound Event Detection based on Duration-Control Model

This chapter describes a novel hybrid approach called duration-controlled long short-term memory (LSTM) for polyphonic Sound Event Detection (SED). It builds upon a state-of-the-art SED method which performs frame-by-frame detection using a bidirectional LSTM recurrent neural network (BLSTM), and incorporates a duration-controlled modeling technique based on a hidden semi-Markov model (HSMM). The proposed approach makes it possible to model the duration of each sound event precisely and to perform sequence-by-sequence detection without the need for thresholding, as is required in conventional frame-by-frame methods. Furthermore, to effectively reduce sound event insertion errors which often occur under noisy conditions, a post-processing step based on binary masks is also introduced. This post-processing step employs a sound activity detection (SAD) network, which determines whether a segment contains only silence or an active sound event of any type, based on an idea inspired by the well-known benefits of voice activity detection in speech recognition systems. Experimental evaluation with the DCASE2016 task 2 dataset are conducted to compare the proposed method with typical conventional methods, such as non-negative matrix factorization (NMF) and standard BLSTM. The proposed method outperforms the conventional methods both in an event-based evaluation, achieving a 75.3% F1 score and a 44.2% error rate, and in a segment-based evaluation, achieving an 81.1% F1 score and a 32.9% error rate, outperforming the best results reported in the DCASE2016 task 2 Challenge. Furthermore, combining the three BLSTM-based methods allows further improvements.

## 4.1 Introduction

The goal of sound event detection (SED) is to detect the beginning and the end of sound events and to label them. SED has great potential for use in many applications, such as retrieval from multimedia databases [9], life-logging [11], activity monitoring [10, 113], environmental context understanding [114], automatic control of devices in smart homes [12], and analysis of noise pollution [115]. Improvements in machine learning techniques have opened new opportunities for progress in these challenging tasks. As a result, SED has been attracting more and more attention, and research in the field is becoming more active. It is notable that several SED challenges have recently been held, such as the CLEAR AED [1], the TRECVID MED [2], and the DCASE Challenge [3–6].

SED can be divided into two main categories, monophonic and polyphonic, which are used in different scenarios. In monophonic SED, the maximum number of simultaneously active sound events is assumed to be one. Because real-world sound events include a wide range of sounds, which vary in their acoustic characteristics, duration, and volume, correctly detecting and identifying such sound events is already difficult, even in the monophonic case. On the other hand, in polyphonic SED, there can be any number of simultaneously active sound events. Polyphonic SED is a more realistic task than monophonic SED because several sound events are likely to occur simultaneously in real-world situations. But it is also even more difficult, due to the overlapping of multiple sound events.

The most typical SED method is to use a hidden Markov model (HMM), where the emission probability distribution is represented by Gaussian mixture models (GMM-HMM), with Mel frequency cepstral coefficients (MFCCs) as features [50, 51]. In the GMM-HMM approach, each sound event, as well as silence between events, is modeled by an HMM, and the maximum likelihood path is determined using the Viterbi algorithm. However, this approach typically achieves only limited performance, and requires heuristics, such as the number of simultaneously active events, to perform polyphonic SED.

Another approach is to use non-negative matrix factorization (NMF) [52–55]. In NMF approaches, a dictionary of basis vectors is learned by decomposing the spectrum of each single-sound event into the product of a basis matrix and an activation matrix, and then combining the basis matrices of all the sound events. The activation matrix for testing is estimated using the combined basis vector dictionary, and is then used either for estimating sound event activations or as a feature vector that is passed on to a classifier. These NMF-based methods can achieve good performance, but they do not take correlations in the time direction into account, instead performing frame-by-frame processing. As a result, the prediction results lack temporal stability so that extensive post-processing is needed. Moreover, the optimal number of bases for each sound event also must be identified.

More recently, methods based on neural networks have been developed, which have also achieved good SED performance [56–62]. A single network is typically trained to solve a multi-label classification problem involving polyphonic SED. Some studies [57, 59, 60, 62] have also utilized recurrent neural networks (RNN), which are able to take into account correlations in the time direction. Although these approaches achieve good performance, they must still perform frame-by-frame detection and they do explicitly model the duration of the output label sequence. Additionally, threshold values for the actual outputs need to be determined carefully to return the best performance.

In this study, a duration-controlled LSTM system which is a hybrid system of a hidden semi-Markov model and a bidirectional long short-term memory RNN (BLSTM-HSMM) is proposed, where output duration is explicitly modeled by an HSMM on top of a BLSTM network. The proposed hybrid system is inspired by the BLSTM-HMM hybrid system used in speech recognition systems [116–118]. In this study, the use of the hybrid system is extended to polyphonic SED and, more generally, to multi-label classification problems. The proposed approach not only allows the implicit capture of various sound event characteristics and correlation information in the time axis, through the use of deep recurrent neural networks with low-level features, but also allows to explicitly model the duration of each sound event through the use of an HSMM. This makes it possible to perform sequence-by-sequence detection without the need for thresholding, as is required in conventional frame-by-frame methods. Furthermore, to effectively reduce sound event insertion errors which are often observed under noisy conditions, a post-processing step based on binary masks is also introduced. This post-processing step employs a sound activity detection (SAD) network, which determines whether a segment contains only silence or an active sound event of any type, based on an idea inspired by the well-known benefits of voice activity detection in speech recognition systems [119–121].

The rest of this chapter is organized as follows: Section 4.2 discusses various types of recurrent neural networks and the concept of long short-term memory. Section 4.3 explains the basics of hidden semi-Markov models. Section 4.4 describes the proposed method in detail. Section 4.5 describes the design of the experimental evaluation and the performance of the proposed method is compared with conventional methods. Finally, this chapter is summarized in Section 4.6.

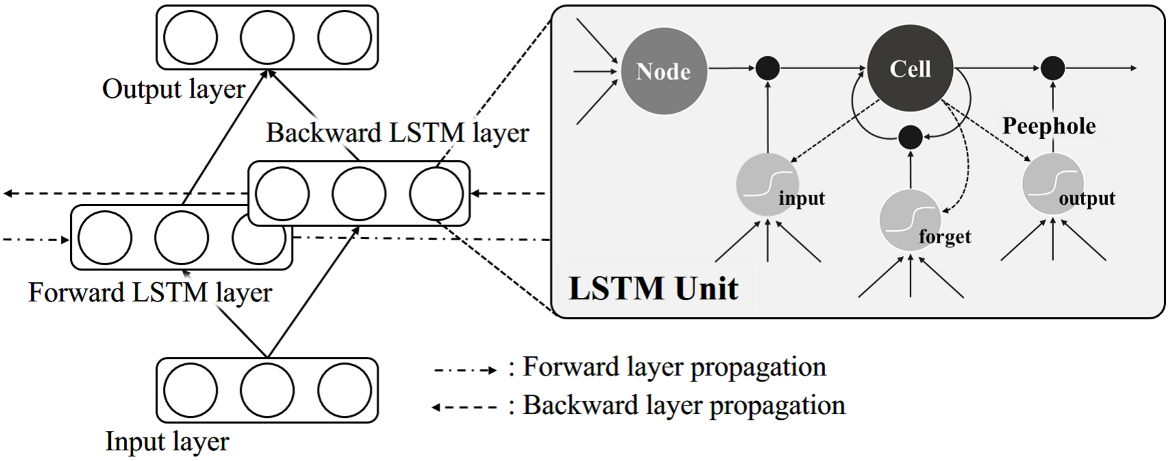


Figure 4.1: *Overview of BLSTM.*

## 4.2 Overview of Recurrent Neural Network Architectures

### 4.2.1 Bidirectional long short-term memory

A bidirectional long short-term memory recurrent neural network (BLSTM) is a layered network with feedback structures from both the previous time step and the following time step, whose layers consist of long short-term memory (LSTM) [122, 123]. Compared with unidirectional structures, the bidirectional structure makes it possible to propagate information not only from the past but also from the future, giving bidirectional networks the ability to exploit the full context of an input sequence. LSTM architectures prevent the so-called *vanishing gradient* problem [124] and allow the memorization of long-term context information. The structure of a BLSTM is illustrated in Fig. 4.1 (for simplicity of presentation, Fig. 4.1 and the following formulations only consider a single hidden layer). As shown in Fig. 4.1, LSTM layers are characterized by a *memory cell* $\mathbf{s}_t$, and three gates: 1) an *input gate* $\mathbf{g}_t^{(I)}$, 2) a *forget gate* $\mathbf{g}_t^{(F)}$, and 3) an *output gate* $\mathbf{g}_t^{(O)}$. Each gate $\mathbf{g}^*$ has a value between 0 and 1. The value 0 means the gate is closed, while the value 1 means the gate is open.

The memory cell memorizes information about the past, the input gate decides whether to pass on the input, and the output gate decides whether to pass on the output. In other words, these gates prevent the propagation of unnecessary signals. The forget gate decides whether to forget the information memorized in the memory cell.

Let us denote a sequence of feature vectors as $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$. In an LSTM layer, the output vector of the LSTM layer $\mathbf{h}_t$ is calculated as follows:

$$\begin{aligned}
\mathbf{g}_t^{(I)} &= \sigma(\mathbf{W}^{(I)}\mathbf{x}_t + \mathbf{W}_r^{(I)}\mathbf{h}_{t-1} + \mathbf{p}^{(I)} \odot \mathbf{s}_{t-1} + \mathbf{b}^{(I)}), && (4.1)\\
\mathbf{g}_t^{(F)} &= \sigma(\mathbf{W}^{(F)}\mathbf{x}_t + \mathbf{W}_r^{(F)}\mathbf{h}_{t-1} + \mathbf{p}^{(F)} \odot \mathbf{s}_{t-1} + \mathbf{b}^{(F)}), && (4.2)\\
\mathbf{s}_t &= \mathbf{g}_t^{(I)} \odot f(\mathbf{W}^{(C)}\mathbf{x}_t + \mathbf{W}_r^{(C)}\mathbf{h}_{t-1} + \mathbf{b}^{(C)}) + \mathbf{g}_t^{(F)} \odot \mathbf{s}_{t-1}, && (4.3)\\
\mathbf{g}_t^{(O)} &= \sigma(\mathbf{W}^{(O)}\mathbf{x}_t + \mathbf{W}_r^{(O)}\mathbf{h}_{t-1} + \mathbf{p}^{(O)} \odot \mathbf{s}_t + \mathbf{b}^{(O)}), && (4.4)\\
\mathbf{h}_t &= \mathbf{g}_t^{(O)} \odot \tanh(\mathbf{s}_t), && (4.5)
\end{aligned}$$

where superscripts $I$, $F$, $O$, and $C$ indicate the input, forget, output gates, and memory cell, respectively, $\odot$ represents point-wise multiplication, $\sigma$ represents a logistic sigmoid function, $f$ represents an activation function, $\mathbf{W}^*$ and $\mathbf{W}_r^*$ denote the input weight matrices and recurrent weight matrices, respectively, $\mathbf{b}^*$ are bias vectors, and $\mathbf{p}^*$ are peephole connection weights. A peephole connection is a connection between a memory cell and a gate, which enables to control the behavior of a gate depending on the state of the memory cell. In a BLSTM layer, the output vector of the forward LSTM layer $\overrightarrow{\mathbf{h}}_t$ and that of the backward LSTM layer $\overleftarrow{\mathbf{h}}_t$ (defined similarly to the forward layer but with all sequences time-reversed) are both calculated using these equations. Finally, the output vector of the output layer $\mathbf{y}_t$ is calculated as follows:

$$\mathbf{y}_t = g\left(\overrightarrow{\mathbf{W}}\overrightarrow{\mathbf{h}}_t + \overleftarrow{\mathbf{W}}\overleftarrow{\mathbf{h}}_t + \mathbf{b}\right), \tag{4.6}$$

where $g$ represents an activation function, $\overrightarrow{\mathbf{W}}$ and $\overleftarrow{\mathbf{W}}$ represent the weight matrix between the output layer and the forward LSTM layer, and between the output layer and the backward

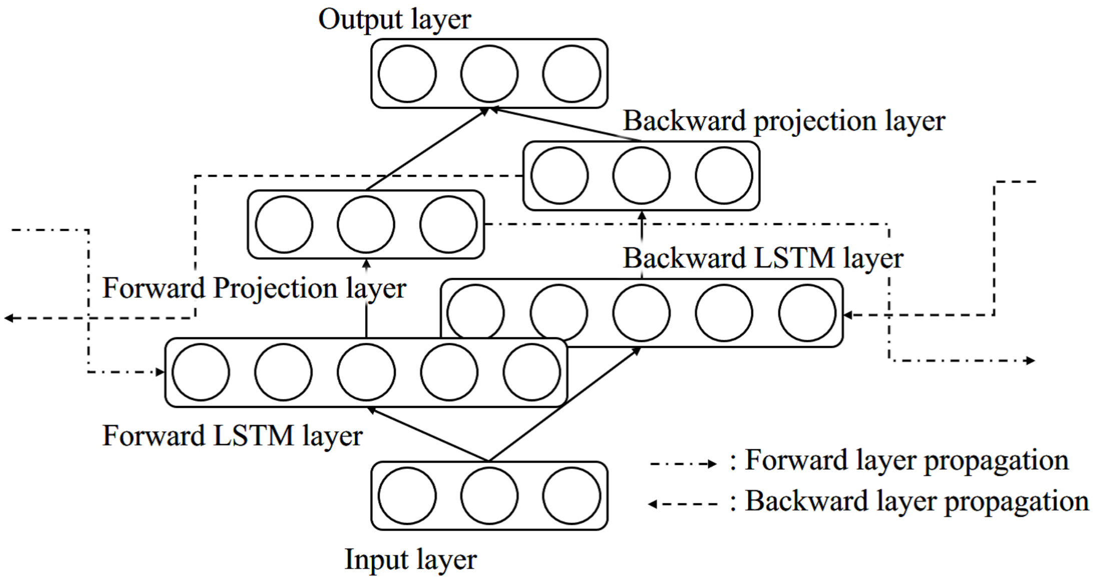


Figure 4.2: *Overview of BLSTM with a projection layer.*

LSTM layer, respectively, while **b** denotes the bias vector of the output layer.

### 4.2.2 Projection layer

In general, it is known that the deep structure of neural networks is what gives them their impressive generalization power. However, building a deep LSTM network with a lot of parameters requires huge memory resources and involves high computational costs. Projection layers have been proposed as a way to address this issue and allow the creation of very deep LSTM networks [116]. The use of projection layers can reduce not only the computational cost but also the effect of overfitting, improving the generalization power. The architecture of a BLSTM with a projection layer is shown in Fig. 4.2. The projection layer, which is a linear transformation layer, is inserted after an LSTM layer, and it outputs feedback to the LSTM layer. With the insertion of a projection layer, the hidden layer output $\mathbf{h}_{t-1}$ in Eqs. (4.1)-(4.4) is replaced with the projection layer output $\mathbf{p}_{t-1}$, and the following equation is added:

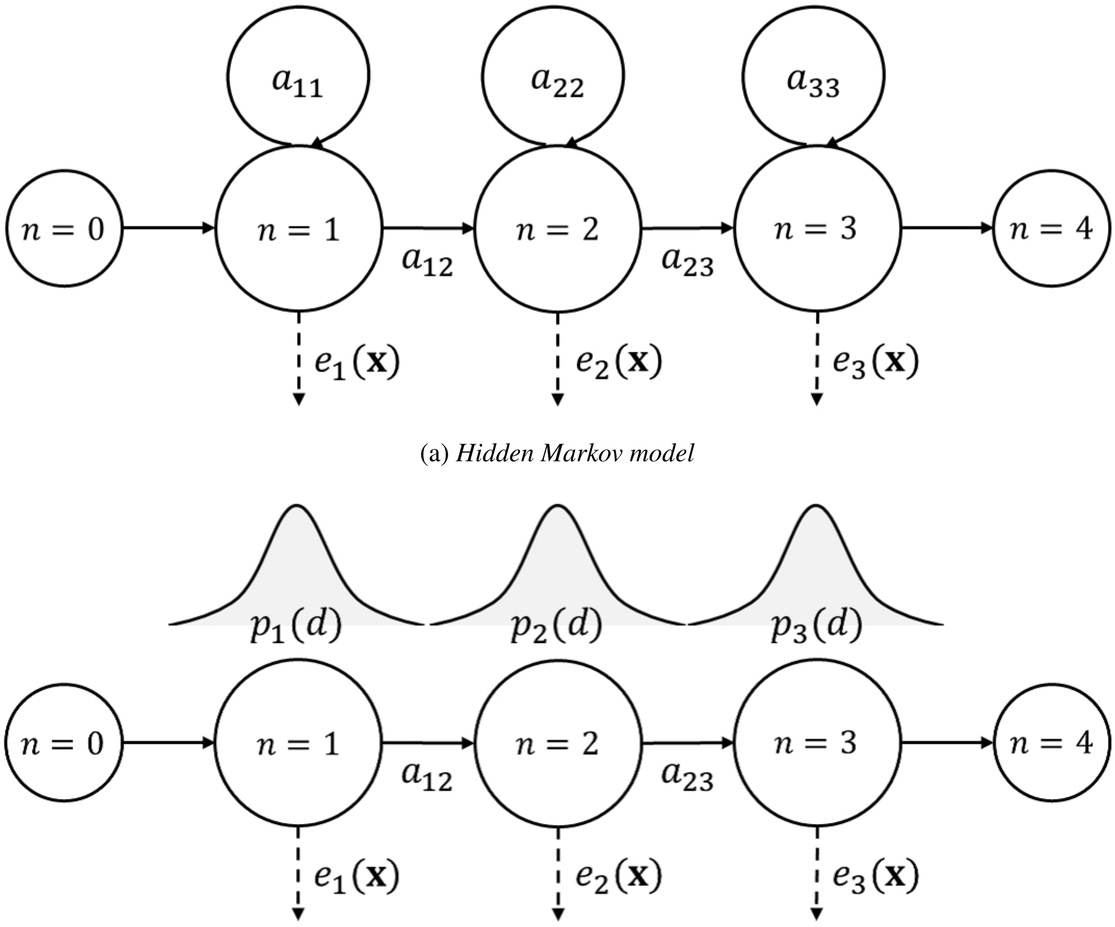
$$\mathbf{p}_t = \mathbf{W}^{(P)}\mathbf{h}_t, \tag{4.7}$$

where $\mathbf{W}^P$ denotes the projection weight matrix.

## 4.3 Overview of Hidden Semi-Markov Model

### 4.3.1 Hidden Markov model

A hidden Markov model (HMM) is a well-known generative model which can deal with variable-length sequential data. To make the extension to the hidden semi-Markov model (HSMM) easier to understand, we first give a brief overview of HMMs. The structure of a typical left-to-right HMM is shown in Fig. 4.3(a). Let us assume that we have sequential observations $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$. Each HMM state $i \in \mathcal{S} = \{1, \ldots, N\}$ has an emission probability distribution $e_i(\mathbf{x})$ and the transition from state $i$ to state $j$ is represented by the transition probability $a_{ij}$. While the emission probability distribution $e_i(\mathbf{x})$ is typically modeled with a Gaussian mixture model (GMM), in a hybrid neural network/HMM model, it is calculated using a pseudo likelihood trick (see Section 4.4.3). The maximum likelihood estimate of a state sequence for an HMM is commonly obtained using the Viterbi algorithm. To perform the Viterbi algorithm, two variables are introduced: the forward variable $\delta_t(i)$, which represents the maximum likelihood that the partial state sequence ends at time $t$ in state $i$, and the back pointer $\psi_t(i)$, which records the corresponding likelihood-maximizing pre-transition state. The forward variable $\delta_0(i)$ is initialized using an initial state probability



(a) *Hidden Markov model*

(b) *Hidden semi-Markov model*

Figure 4.3: *Structural difference between an HMM and HSMM.*

**Algorithm 1** HMM Viterbi algorithm

```
Initialization:
$\delta_0(i) = \pi_i,\ i \in \mathcal{S}$
$\psi_0(i) = 0,\ i \in \mathcal{S}$
Inference:
for $t = 1$ to $T$ do
    for $j = 1$ to $N$ do
        $\delta_t(j) = \max_{i \in \mathcal{S}} \{\delta_{t-1}(i) a_{ij} e_j(\mathbf{x}_t)\}$
        $\psi_t(j) = \arg\max_{i \in \mathcal{S}} \{\delta_{t-1}(i) a_{ij} e_j(\mathbf{x}_t)\}$
    end for
end for
Termination:
$P^* = \max_{i \in \mathcal{S}} \{\delta_T(i)\}$
$s_T^* = \arg\max_{i \in \mathcal{S}} \{\delta_T(i)\}$
Traceback:
for $t = T - 1$ to $1$ do
    $s_t^* = \psi_{t+1}(s_{t+1}^*)$
end for
```

$\pi_i$, and the back pointer $\psi_0(i)$ is initialized as 0. These are calculated recursively as follows:

$$\delta_t(j) = \max_{i \in \mathcal{S}} \{\delta_{t-1}(i) a_{ij} e_j(\mathbf{x}_t)\}, \tag{4.8}$$

$$\psi_t(j) = \arg\max_{i \in \mathcal{S}} \{\delta_{t-1}(i) a_{ij} e_j(\mathbf{x}_t)\}. \tag{4.9}$$

After the recursive calculation, the maximum likelihood $P^*$ and maximum likelihood state $s_T^*$ are calculated as follows:

$$P^* = \max_{i \in \mathcal{S}} \{\delta_T(i)\}, \tag{4.10}$$

$$s_T^* = \arg\max_{i \in \mathcal{S}} \{\delta_T(i)\}. \tag{4.11}$$

Now, we can calculate the maximum likelihood path $\{s_1^*, s_2^*, \ldots, s_T^*\}$ using the following equation:

$$s_t^* = \psi_{t+1}(s_{t+1}^*), \tag{4.12}$$

starting at the maximum likelihood state $s_T^*$ at time $T$ and moving backwards. The entire process of maximum likelihood estimation in HMMs using the Viterbi algorithm is shown in Algorithm 1.

### 4.3.2 Hidden semi-Markov model

One major problem with HMMs is that they are limited in their ability to represent state duration. In HMMs, state duration probabilities are implicitly represented by state transition probabilities, therefore, HMM state duration probabilities inherently decrease exponentially with time. However, duration probabilities in real data do not necessarily follow an exponentially decreasing distribution. Consequently, the representation of duration in HMMs may be inappropriate and cause a discrepancy between the model and the data. This will be especially apparent in SED, where it is necessary to deal with various types of sounds with various durations.

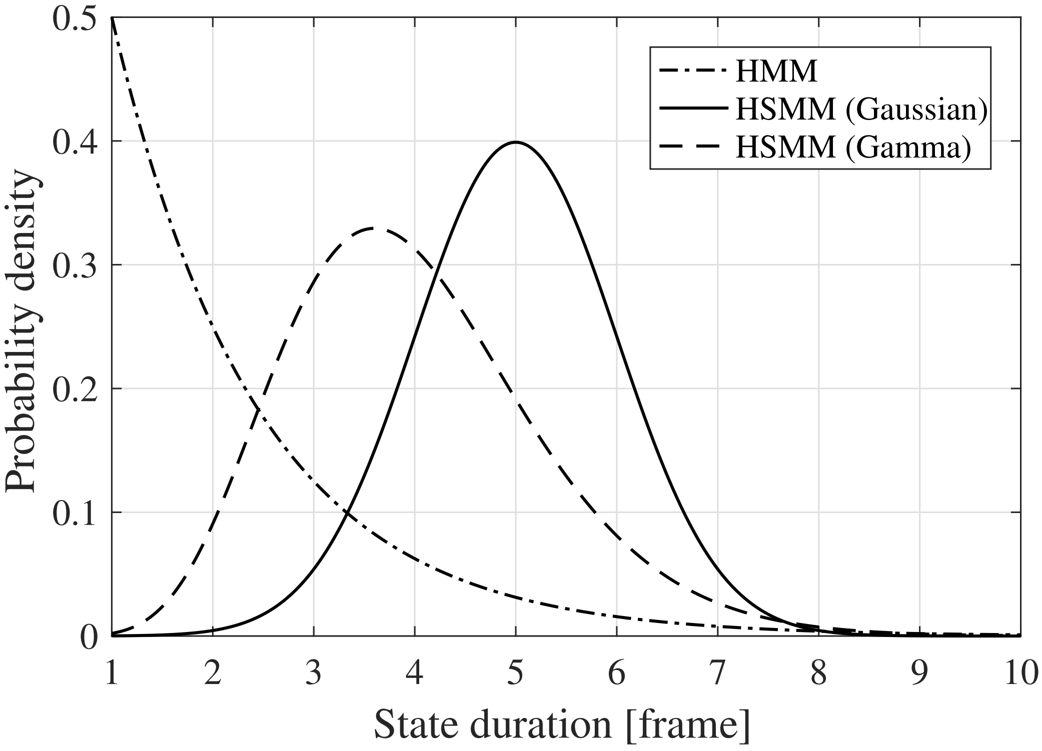


Figure 4.4: *Difference in duration probability.*

One solution to this problem is to use hidden semi-Markov models [125, 126]. The structure of an HSMM is shown in Fig. 4.3(b), and the difference in state duration probability between an HMM and an HSMM is shown in Fig. 4.4. In HSMMs, duration $d \in \mathcal{D} = \{1, 2, \ldots, D\}$ at state $j$ is explicitly modeled by a probability distribution $p_j(d)$. The parameters of $p_j(d)$ are estimated using maximum likelihood estimation, and the maximum duration $D$ is decided based on the mean and variance of $p_j(d)$. The Viterbi algorithm can be extended to the case of HSMMs by modifying the definitions and recurrence formulas

**Algorithm 2** HSMM Viterbi algorithm

```
Initialization:
$\delta_d(i,d) = \pi_i p_i(d) e_i(\mathbf{x}_{1:d}),\ i \in \mathcal{S}, d \in \mathcal{D}$
$\boldsymbol{\psi}_d(i,d) = (0,0,0)\ i \in \mathcal{S}, d \in \mathcal{D}$
Inference:
for $t = 1$ to $T$ do
  for $j = 1$ to $N$ do
    for $d = 1$ to $D$ do
      $\delta_t(j,d) = \max_{i \in \mathcal{S}, d' \in \mathcal{D}} \{\delta_{t-d}(i,d') a_{ij} p_j(d) e_j(\mathbf{x}_{t-d+1:t})\}$
      $(s^*, d^*) = \arg\max_{i \in \mathcal{S}, d' \in \mathcal{D}} \{\delta_{t-d}(i,d') a_{ij} p_j(d) e_j(\mathbf{x}_{t-d+1:t})\}$
      $\boldsymbol{\psi}_t(j,d) = (t-d, s^*, d^*)$
    end for
  end for
end for
Termination:
$P^* = \max_{i \in \mathcal{S}, d \in \mathcal{D}} \{\delta_T(i,d)\}$
$(s_1^*, d_1^*) = \arg\max_{i \in \mathcal{S}, d \in \mathcal{D}} \{\delta_T(i,d)\}$
Traceback:
$t_1 = T, n = 1$
while $t_n > 1$ do
  $n \leftarrow n + 1$
  $(t_n, s_n^*, d_n^*) = \boldsymbol{\psi}_{t_{n-1}}(s_{n-1}^*, d_{n-1}^*)$
end while
```

of the forward variable $\delta$ and the back pointer $\psi$ as follows:

$$\delta_t(j,d) = \max_{i\in\mathcal{S},d'\in\mathcal{D}} \{\delta_{t-d}(i,d')a_{ij}p_j(d)e_j(\mathbf{x}_{t-d+1:t})\}, \tag{4.13}$$

$$\boldsymbol{\psi}_t(j,d) = (t-d, s^*, d^*), \tag{4.14}$$

where $s^*$, $d^*$ and $t-d$ represent the previous state, its duration, and its end time, respectively, and $s^*$ and $d^*$ are calculated using the following equation:

$$(s^*, d^*) = \arg\max_{i\in\mathcal{S},d'\in\mathcal{D}} \{\delta_{t-d}(i,d')a_{ij}p_j(d)e_j(\mathbf{x}_{t-d+1:t})\}, \tag{4.15}$$

where the value of $e_j(\mathbf{x}_{t-d+1:t})$ is computed as follows:

$$e_j(\mathbf{x}_{t-d+1:t}) = \prod_{\tau=t-d+1}^{t} e_j(x_\tau). \tag{4.16}$$

After the recursive calculation, the maximum likelihood $P^*$, maximum likelihood state $s_1^*$ and its duration $d_1^*$ are calculated as follows:

$$P^* = \max_{i\in\mathcal{S},d\in\mathcal{D}} \{\delta_T(i,d)\}, \tag{4.17}$$

$$(s_1^*, d_1^*) = \arg\max_{i\in\mathcal{S},d\in\mathcal{D}} \{\delta_T(i,d)\}. \tag{4.18}$$

Finally, we can obtain the maximum likelihood path $\{s_N^*, s_{N-1}^*, \ldots, s_1^*\}$ and its duration $\{d_N^*, d_{N-1}^*, \ldots, d_1^*\}$ by applying the following equation recursively:

$$(t_n, s_n^*, d_n^*) = \boldsymbol{\psi}_{t_{n-1}}(s_{n-1}^*, d_{n-1}^*) \tag{4.19}$$

The entire HSMM Viterbi algorithm procedure is shown in Algorithm 2. Note that the ability of the HSMM to explicitly model the duration of each state by introducing the duration probability distribution $p_j(d)$ comes at the expense of an increase in the computational cost of the Viterbi algorithm from $O(NT)$ to $O(NTD)$, as compared with an HMM.

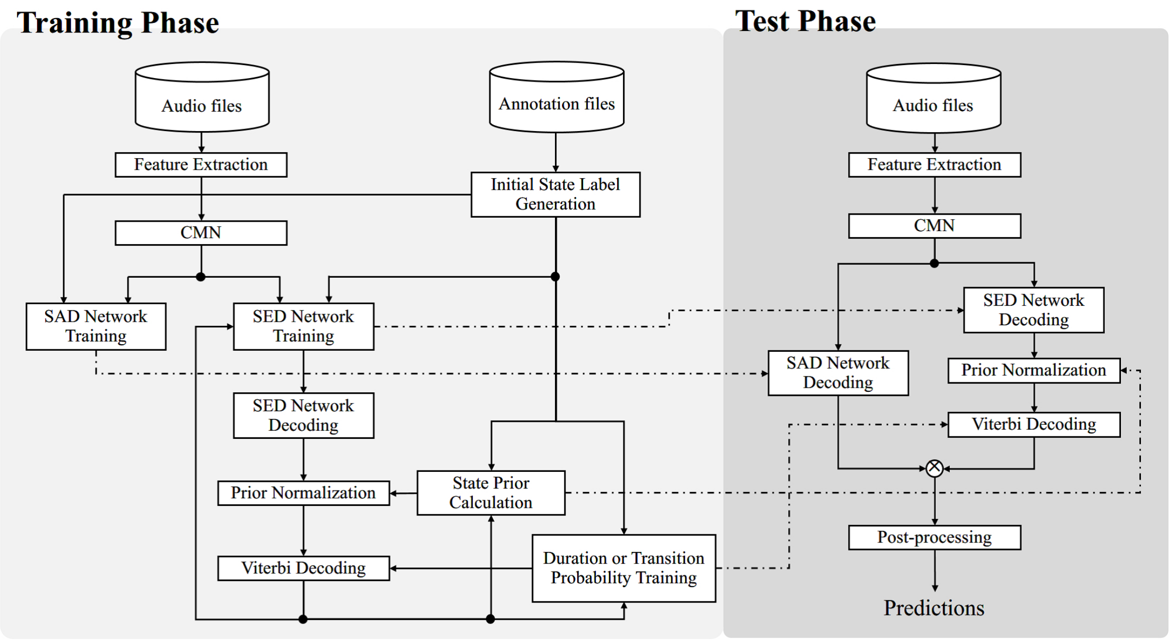


Figure 4.5: *System overview.*

## 4.4 Proposed Method

### 4.4.1 System overview

An overview of the proposed system, separated into training and test phases, is shown in Fig. 4.5. In the training phase, the feature vectors are extracted and cepstral mean normalization (CMN) is performed for each training audio sample (see Section 4.4.2). Each active event is divided into three segments of equal length in order to assign left-to-right states, thus obtaining initial state labels for the HMM (or HSMM). Using the feature vectors and state labels, a sound event detection (SED) network (see Section 4.4.3), and a sound activity detection (SAD) network (see Section 4.4.4) are trained. The labels are also utilized for calculating the priors of HMM and HSMM states, as well as for training the transition probability of the HMM or the duration probability distribution of the HSMM (see Section 4.4.3).

After training, the Viterbi decoding is performed to update the state labels for the training data, and then the training of the SED network and the transition (or duration) probabilities is repeated several times.

In the test phase, the feature vectors are extracted from an input audio sample and CMN is performed. The feature vectors are used as inputs for both the SED network and the SAD network. The SED network estimates each state posterior, while the SAD network estimates a binary mask which indicates global sound event activity, i.e., whether one or more sound events of any types are active in a given segment. Finally, this binary mask is applied to the activation of each sound event obtained using Viterbi decoding (see Section 4.4.4), and post-processing is performed (see Section 4.4.5).

### 4.4.2 Feature extraction

The input signal is divided into 25 ms windows with 40% overlap, and 100 log mel-filterbank features are then calculated for each window. More bands than usual are used since the resolution of high frequency components is important for SED. Next, cepstral mean normalization is performed for each piece of training data, obtaining an input feature vector $\mathbf{x}_t$ at each frame $t$. These operations are performed using HTK [127].

### 4.4.3 Hybrid model

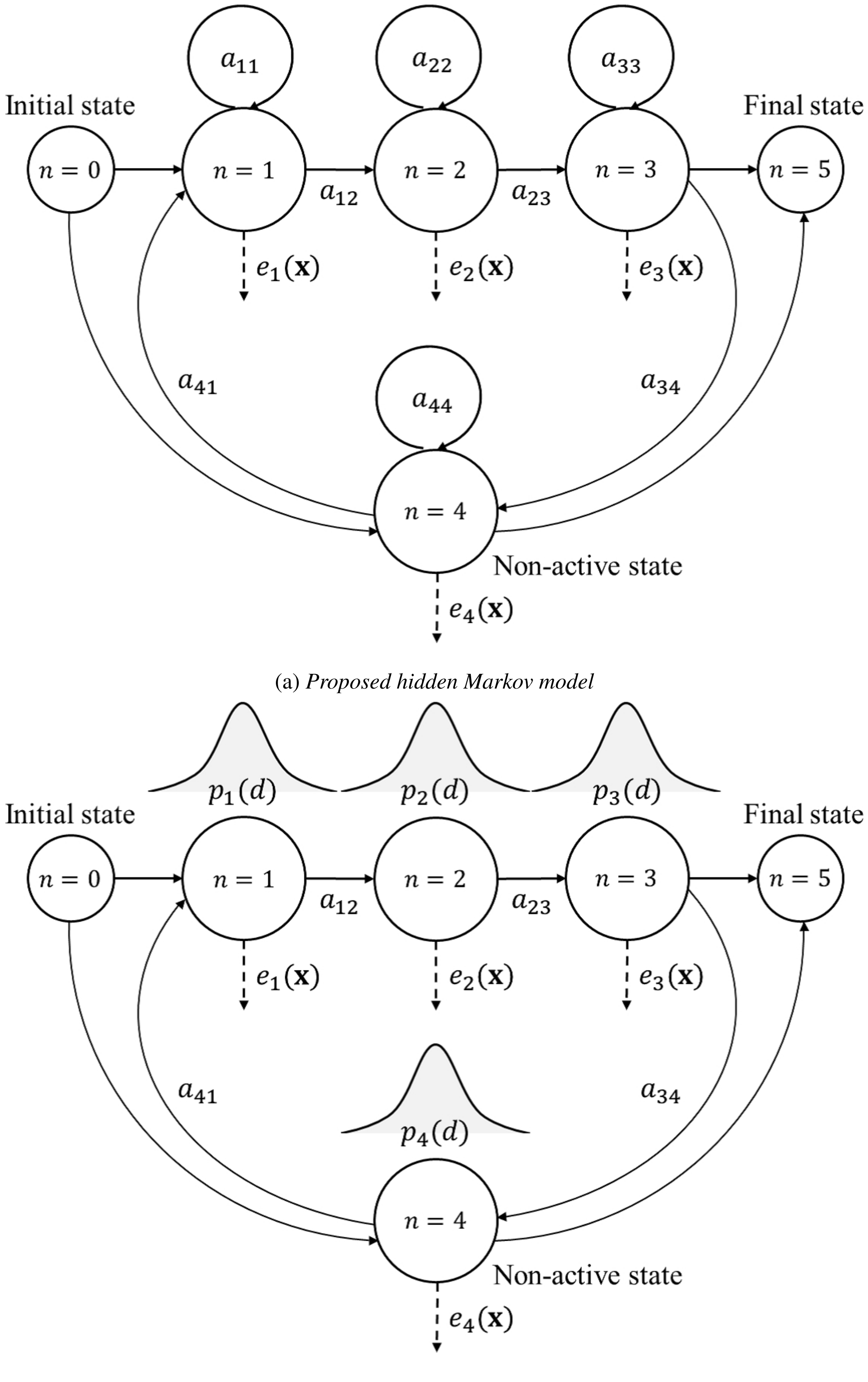
Two hybrid systems, BLSTM-HMM and BLSTM-HSMM, are used to capture sound-event-dependent temporal structures explicitly and also to perform sequence-by-sequence detection without the thresholding that is necessary when using conventional frame-by-frame methods. While the BLSTM-HMM implicitly models the duration of each sound event via its transition probabilities, the BLSTM-HSMM can explicitly do so via its duration probabilities. The hybrid neural network/HMM framework, which is generally used to handle multi-class classification problems, is extended to handle multi-label classification problems for polyphonic SED. To do this, a three-state, left-to-right HMM (or HSMM) with a non-active state is built for each sound event. The structures of the HMM and HSMM are shown in Figs. 4.6(a) and 4.6(b), respectively, where $n = 0$, $n = 5$ and $n = 4$ represent the *initial state*, *final state*, and *non-active state*, respectively. Note that the non-active state only pertains to the absence of activity for a particular sound event, and does not indicate whether other sound events are active or not.

For the HMM, the transition probabilities are learned from the sequences of state labels, where the number of transitions from state $i$ to state $j$ is simply calculated and it is normalized to meet the definition of a probability. On the other hand, for the HSMM, the transition probabilities of the left-to-right states are fixed at 0.99 (the remaining 0.01 corresponding to the self-loop), and that of the non-active state is fixed at 0.01 (with the remaining 0.99 corresponding to the self-loop). These transition probabilities were determined through preliminary experiments. Duration is represented using the gamma distribution in the three left-to-right states and a uniform distribution in the non-active state. The duration probability $p_{c,j}(d)$ for state $j$ of the HSMM for event $c$ is defined as follows:

$$p_{c,j}(d) = \begin{cases} d^{k_{c,j}-1} \frac{\exp(-\theta_{c,j} d)}{\theta_{c,j}^{k_{c,j}} \Gamma(k_{c,j})} & (j = 1, 2, 3) \\ \frac{1}{D_c} & (j = 4) \end{cases}, \tag{4.20}$$

where $k_{c,j}$ and $\theta_{c,j}$ respectively denote the shape and scale parameters of the gamma distribution, obtained through maximum likelihood estimation using the state labels, and $D_c$ is the maximum duration length of the three left-to-right states for event $c$. In this study, $D_c$ is determined as follows:

$$D_c = \max_{j \in \{1,2,3\}} \left\{ \mu_{c,j} + 3\sigma_{c,j} \right\}, \tag{4.21}$$



(a) *Proposed hidden Markov model*

(b) *Proposed hidden semi-Markov model*

Figure 4.6: *Difference between proposed HMM and HSMM.*

where $\mu_{c,j}$ and $\sigma_{c,j}$ respectively denote the mean and standard deviations of the duration of state $j$ of event $c$.

In the hybrid model, the network is used to calculate the state posterior $P(s_{c,t} = j|\mathbf{x}_t)$, where $c \in \{1, 2, \ldots, C\}$ denotes the sound event index, $j \in \{1, 2, \ldots, N\}$ the index of states except for the initial and final states, and $s_{c,t}$ represents the state of event $c$ at time $t$. The emission probability $e_{c,j}(\mathbf{x}_t)$ for state $j$ of event $c$ can be obtained from the state posterior using Bayes' theorem as follows:

$$e_{c,j}(\mathbf{x}_t) = P(\mathbf{x}_t|s_{c,t} = j) = \frac{P(s_{c,t} = j|\mathbf{x}_t)P(\mathbf{x}_t)}{P(s_{c,t} = j)}, \tag{4.22}$$

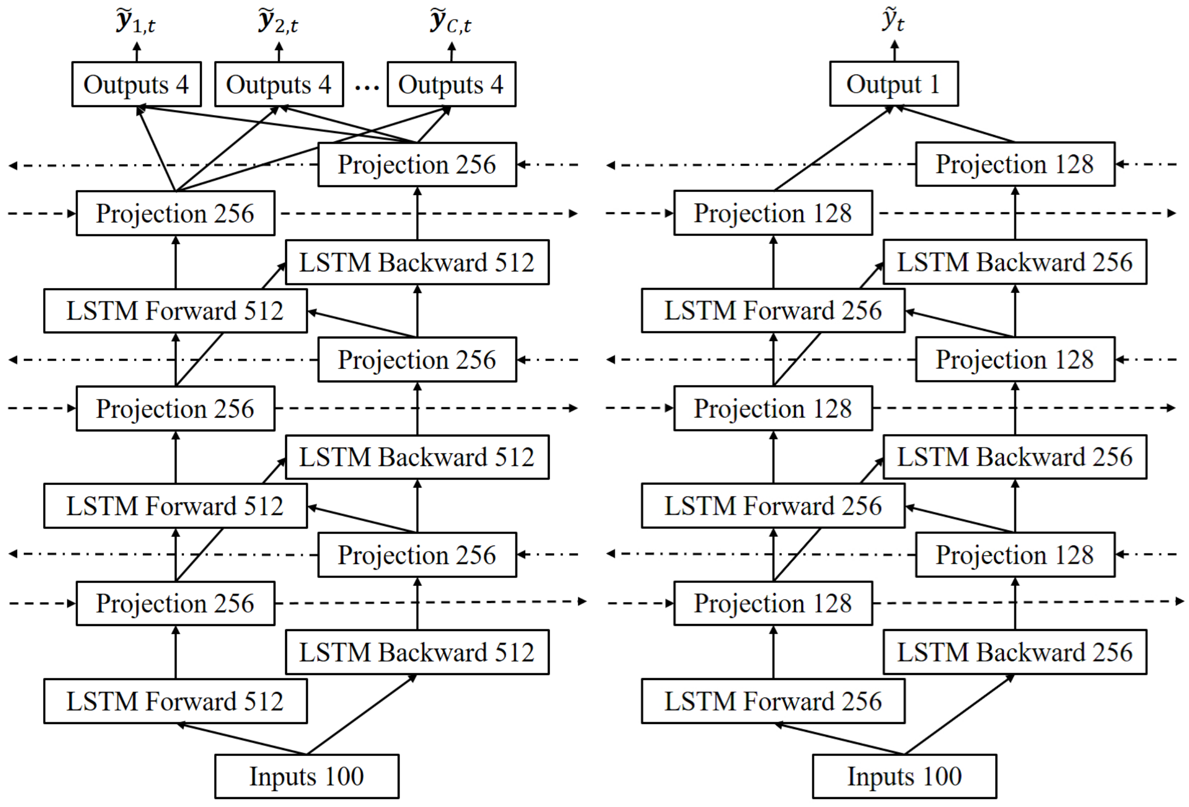
where $P(s_{c,t})$ represents the prior of HMM (or HSMM) states. Note that the factor $P(\mathbf{x}_t)$ is irrelevant in the Viterbi computations. The structure of the proposed network is shown in Fig. 4.7(a), where $\tilde{\mathbf{y}}_{c,t}$ represents the state posterior of event $c$ at time $t$. This network has three hidden layers, each consisting of a BLSTM layer with 1,024 nodes and a projection layer with 512 nodes, and an output layer with $C \times N$ nodes. These network structures are determined through preliminary experiments. A softmax operation is used to ensure that the values of posterior $P(s_{c,t}|\mathbf{x}_t)$ sum to one for each sound event $c$ in frame $t$, as follows:

$$P(s_{c,t} = j|\mathbf{x}_t) = \frac{\exp(a_{c,j,t})}{\sum_{j'=1}^{N} \exp(a_{c,j',t})}, \tag{4.23}$$

where $a$ represents the activation of the output layer node. The network is optimized using back-propagation through time (BPTT) [128] with Adam [106] and dropout [107] using cross-entropy as shown in the following *multi-class, multi-label* objective function:

$$E(\mathbf{\Theta}) = -\sum_{c=1}^{C}\sum_{j=1}^{N}\sum_{t=1}^{T} y_{c,j,t} \ln(P(s_{c,t} = j|\mathbf{x}_t)), \tag{4.24}$$

where $\mathbf{\Theta}$ represents the set of network parameters, and $y_{c,j,t}$ is the state label. This objective function can be seen as a kind of multi-task learning [129] of multi-class classification problems. Multi-task learning is a technique to solve a task with additional tasks using a



(a) *SED network* (b) *SAD network*

Figure 4.7: *Proposed network structures.*

shared input representation, and the effectiveness has been confirmed in various fields such as speech recognition [130], and natural language processing [131]. Note that this objective function is not the same as the multi-class objective function in a conventional DNN-HMM because the SED network estimates state posteriors with respect to each sound event HMM (or HSMM), not the state posteriors of states of all HMMs (or HSMMs). State prior $P(s_{c,t})$ is calculated by counting the number of occurrences of each state using the training data labels. However, in this study, since the synthetic training data does not represent actual sound event occurrences, the prior obtained from occurrences of each state has to be made less sensitive. Therefore, $P(s_{c,t})$ is smoothed as follows:

$$\hat{P}(s_{c,t}) = P(s_{c,t})^{\alpha}, \tag{4.25}$$

where $\alpha$ is a smoothing coefficient. In this study, $\alpha$ is set to 0.01. Finally, the state emission probability is calculated using Eq. (4.22) and the Viterbi algorithm is performed for each HMM (or HSMM) to obtain the maximum likelihood path as shown in Section 4.3.

### 4.4.4 SAD network

A common problem when performing polyphonic SED under noisy conditions is a decrease in performance due to insertion errors, which occur when background noise is mistaken for sound events, even though there are no actual sound events in that segment. To solve this problem, the use of binary masking with a sound activity detection (SAD) network is proposed. The SAD network identifies segments in which there is sound event activity of any type, similarly to voice activity detection (VAD) in the field of speech recognition [119–121]. In this study, the network shown in Fig. 4.7(b) is trained. This network has three hidden layers, each consisting of a BLSTM layer with 512 nodes, a projection layer with 256 nodes, and an output layer with a single node. The structure of the SAD network is almost the same as that of the SED network, however, the SAD network does not differentiate between the types of the sound events, and therefore it tends to concentrate on background noise. The SAD network, which performs a simple binary classification, is optimized using BPTT with Adam and dropout under the following sigmoid cross-entropy objective:

$$E(\mathbf{\Theta}) = -\sum_{t=1}^{T} \{y_t \ln(\tilde{y}_t) + (1 - y_t) \ln(1 - \tilde{y}_t)\}, \tag{4.26}$$

where $y_t$ represents the reference data indicating the presence or absence of sound events and $\tilde{y}_t$ is the SAD network output. A threshold of 0.5 is used to convert SAD network outputs into a binary mask $\mathbf{M}$, and it is applied to the activations $\mathbf{A}_c$ of each sound event $c$ predicted by the BLSTM-HMM (or BLSTM-HSMM), as follows:

$$\tilde{\mathbf{A}}_c = \mathbf{M} \odot \mathbf{A}_c, \tag{4.27}$$

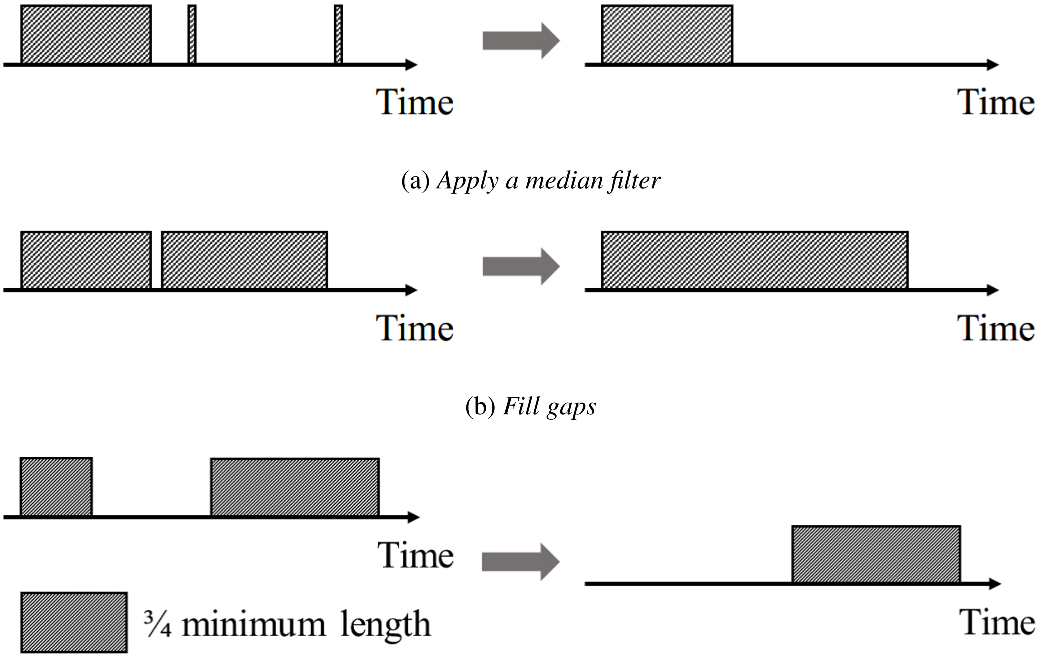
where both $\mathbf{A}_c$ and $\mathbf{M}$ are a binary vector of length $T$. Note that the same binary mask $\mathbf{M}$ is applied to the activations of each sound event, and that the binary mask only has an effect on the insertion of sound events, not on the substitution or deletion of sound events.

### 4.4.5 Post-processing

In order to smooth the output sequence, three kinds of post-processing are performed:

1. Apply a median filter with a predetermined filter span;
2. Fill gaps which are shorter than a predetermined number;
3. Remove events whose duration is shorter than a predetermined length.

An outline of each post-processing step is illustrated in Fig. 4.8. Based on preliminary experiments, the median filter span is set to 150 ms (15 frames), the acceptable gap length is set to 0.1 s (10 frames), and the minimum duration threshold length for each sound event is set to 3/4 of the minimum duration for that sound event as calculated from the training data.



(a) *Apply a median filter*

(b) *Fill gaps*

(c) *Remove short duration events*

Figure 4.8: *Diagram of each post-processing step.*

## 4.5 Experimental Evaluation

To evaluate the proposed method, the DCASE2016 task 2 dataset [4] was used. The dataset includes a training set consisting of 20 clean audio files per event class, a development set consisting of 18 synthesized audio files of 120 seconds in length, and an evaluation set consisting of 54 synthesized audio files of the same length as the development set files. The 54 files of the evaluation set consist of 18 audio files with different content each synthesized under three different signal-to-noise ratio (SNR) conditions: −6 dB, 0 dB, and 6 dB. Out of 54 files, 27 are monophonic, and the rest are polyphonic. The number of sound event classes in the dataset is 11. The evaluation set is synthesized using unknown samples, but the development set is synthesized using the training set, making it a closed condition development set.

For this study, the training data was further split, keeping 75% of the samples to build the training set, and holding out the rest in order to build an open condition development set, which was lacking in the original dataset. By open condition development set, a development set was referred here to that is based on data not included in the (newly defined) training set. Thus 5 samples were randomly selected (out of 20) per event from the training set and 18 samples of 120 seconds in length are generated, similar to the original DCASE2016 task 2 development set. These generated samples are used as development data to check performance under open conditions. The remaining 15 samples per class were used to build the original training data. Instead of simply using the original training data, which is too small for training an RNN with sufficient depth, the training data was augmented by synthetically generating training data using the clean samples and background noise as follows:

1. Generate a silence signal of a predetermined length;
2. Randomly select a sound event sample;

3. Add the selected sound event to the generated silence signal at a randomly selected location;
4. Repeat steps 2 and 3 a predetermined number of times;
5. Add a background noise signal at a predetermined SNR.

### 4.5.1 Experimental conditions

The signal length was set to 4 seconds, the total number of events was set to a value from 3 to 5, the number of simultaneous events was set to a value from 1 to 5 (1 corresponding to the monophonic case), and SNR was set to a value from −9 dB to +9 dB. Then 100,000 audio samples were synthesized, for a total length of 111 hours. Both *event-based* (onset-only) and *segment-based* evaluation were conducted, and F1 scores (F1) and error rates (ER) were utilized as the evaluation criteria. Event-based evaluation considers true positives, false positives and false negatives with respect to event instances, while segment-based evaluation is done in a fixed time grid, using segments of one second length to compare the ground truth and the system output (see [132] for more details). The proposed hybrid system was built using the following procedure:

1. Divide an active event into three segments of equal length in order to assign left-to-right state labels;
2. Train the SED network using these state labels as supervised data;
3. Calculate the prior using these state labels;
4. (For HMM) Train the transition probability using these state labels by Viterbi training; (For HSMM) Train the duration probability distribution using the maximum likelihood estimation;
5. Calculate the maximum likelihood path with the Viterbi algorithm;

Table 4.1: *Experimental conditions.*

| | |
|---|---|
| Sampling rate | 44,100 Hz |
| Window size | 25 ms |
| Shift size | 10 ms |
| # training data | 4 s × 100k samples |
| # development data | 120 s × 18 samples |
| # evaluation data | 120 s × 54 samples |
| # sound event classes | 11 |
| Learning rate | 0.001 |
| Initial scale | 0.001 |
| Gradient clipping norm | 5 |
| Batch size | 128 |
| Time steps | 400 |
| Optimization method | Adam [106] |

6. Use the maximum likelihood paths as new state labels;
7. Repeat steps 2-6.

In this study, when calculating the maximum likelihood path, the alignment of non-active states was fixed, i.e., event-active HMM states were only aligned because we know the perfect alignment of non-active states due to the use of synthetic data. When training the networks, the objective functions on the development set were monitored at every epoch and training was stopped using an *early stopping* strategy. All networks were trained using the open source toolkit TensorFlow [133] with a single GPU (Nvidia GTX Titan X). Details of the experimental conditions are shown in Table 4.1.

Table 4.2: *Experimental results.*

| | Event-based (dev. / eval.) | | Segment-based (dev. / eval.) | |
|---|---|---|---|---|
| Model | F1 score [%] | Error rate [%] | F1 score [%] | Error rate [%] |
| NMF (Baseline) | 31.0 / 24.2 | 148.0 / 168.5 | 43.7 / 37.0 | 77.2 / 89.3 |
| BLSTM | 69.9 / 60.1 | 73.2 / 91.2 | 87.2 / 77.1 | 25.8 / 44.4 |
| + post-processing | 81.5 / 71.2 | 35.7 / 50.9 | 89.3 / 79.0 | 20.3 / 36.8 |
| + SAD binary masking | 82.2 / 73.7 | 34.0 / 45.6 | 89.5 / 79.9 | 19.8 / 34.3 |
| BLSTM-HMM | 80.0 / 71.0 | 38.6 / 55.1 | 88.8 / 79.6 | 20.4 / 37.4 |
| + post-processing | 81.3 / 71.7 | 35.2 / 52.3 | 89.1 / 79.5 | 19.4 / 36.7 |
| + SAD binary masking | 82.6 / 74.9 | 32.7 / 44.7 | 89.7 / 80.5 | 18.3 / 33.8 |
| BLSTM-HSMM | 82.3 / 71.9 | 34.7 / 51.7 | 91.3 / 79.8 | 16.7 / 37.0 |
| + post-processing | 82.3 / 72.1 | 34.4 / 51.4 | 91.1 / 79.7 | 16.7 / 37.0 |
| + SAD binary masking | **85.0 / 75.3** | **28.9 / 44.2** | **91.5 / 81.1** | **15.8 / 32.9** |
| I. Choi *et al.* [61] | – / 67.1 | – / 61.8 | – / 78.7 | – / 36.7 |
| T. Komatsu *et al.* [55] | – / 73.8 | – / 46.2 | – / 80.2 | – / 33.1 |

## 4.5.2 Experimental results

To confirm the performance of the proposed method, it was compared with two conventional methods, NMF (DCASE2016 task 2 baseline) [4] and standard BLSTM [57]. The NMF was trained with 15 clean samples per class using the DCASE2016 task 2 baseline script [4]. The BLSTM had the same network structure as the one shown in Fig. 4.7(a), with the exception that the output layer was replaced with $C$ nodes with sigmoid activation functions, with one node for each sound event. Each node's output $y_c \in [0, 1]$ was binarized to determine event activity. The threshold was set to 0.5, i.e., sound event $c$ is considered to be active if $y_c > 0.5$, and inactive otherwise.

Experimental results are shown in Table 4.2. First, we focus on the differences in the performance of each system. The proposed system (a BLSTM-HSMM with post-processing and SAD binary masking) achieved the best performance of all of the methods for both event-based and segment-based evaluation criteria, and also outperformed all of the methods submitted to the DCASE2016 task 2 Challenge [4]. Note that the conventional systems [55, 61] were trained using the whole training set, whereas the systems are trained using only part of the training set in order to hold out data for the preparation of an open development set. From these results we can see that it is important for polyphonic SED methods to take the duration of sound events into account, especially by explicitly modeling event duration.

Next, we focus on the effect of post-processing. For the BLSTM method, we can confirm that post-processing was clearly effective. However, this could not always be confirmed for the proposed hybrid models: this could be expected, because the prediction results are already smoothed as a result of the use of HMM or HSMM. In other words, the use of a dynamic modeling technique such as HMM or HSMM can effectively smooth the output sequence, and therefore, it can alleviate the need for post-processing.

Finally, we focus on the effect of SAD network binary masking. The results confirmed the effectiveness of the proposed SAD masking method when used with all methods, especially for reducing the error rate on the evaluation set. This is because there is more background noise in the evaluation set than in the development set, leading to many insertion errors. (Note that the SAD network by itself achieved an F1 score of 97.7% for the development set and 94.8% for the evaluation set in frame-wise detection.) It is interesting to note that even though almost the same network architecture and exactly the same data were used, an improvement in performance could be obtained by using a slightly different objective function between the SED and SAD networks. This is because using a simple objective function for the SAD network makes it easy to train the network, and combining different tendency models has a performance effect similar to a model ensemble technique. This results also imply that multi-task learning can be used effectively, i.e., the objective function

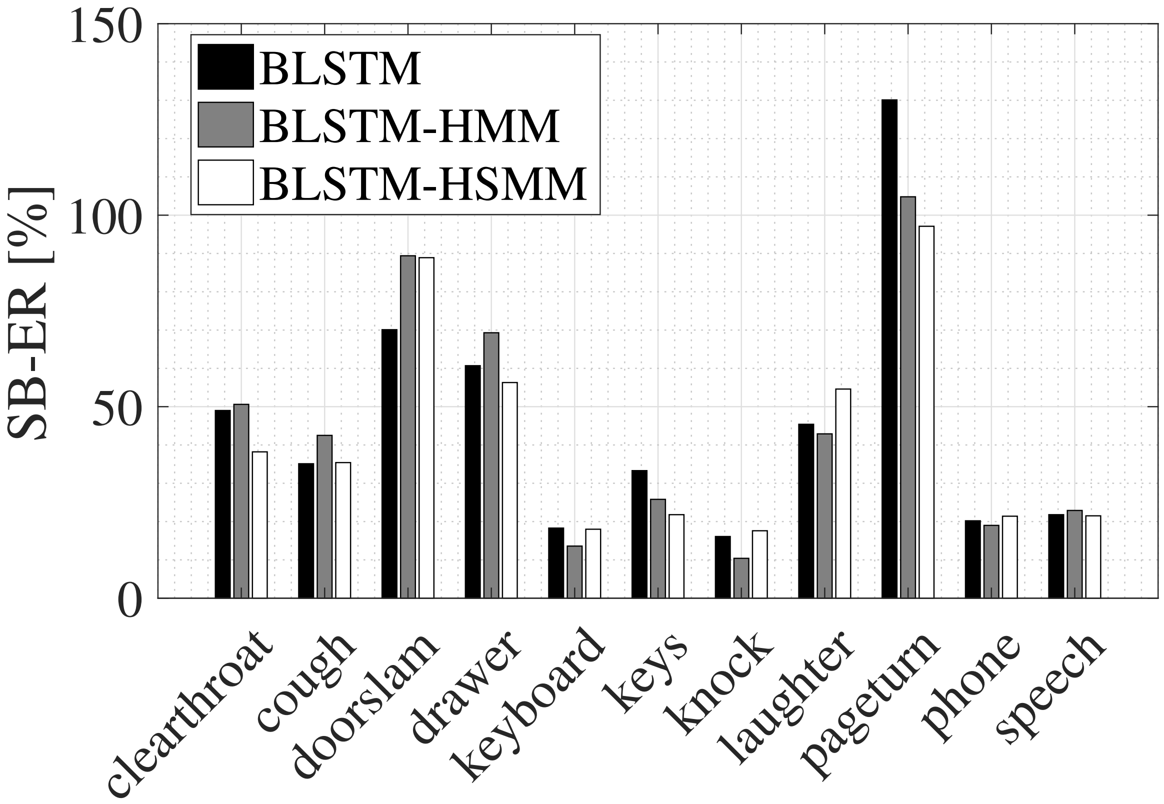


Figure 4.9: *Class-wise segment-based error rates.*

of an SAD network can be utilized as a regularization term in the objective function of an SED network. This will be investigated in future work.

### 4.5.3 Analysis

In this section, the details of the performance of the BLSTM-based methods are discussed. Class-wise segment-based error rates are shown in Fig. 4.9, which displays the rates after both post-processing and SAD binary masking have been applied. Focusing first on differences in performance for various sound events, we note that detection performance for *doorslam*, *drawer*, and *pageturn* were very low. This is because the volume of the *doorslam*

Table 4.3: *Performance under different SNR conditions.*

| | Segment-based (BLSTM / HMM / HSMM) | |
|---|---|---|
| SNR [dB] | F1 score [%] | Error rate [%] |
| 6 | 81.2 / 80.8 / 82.9 | 32.1 / 32.6 / 28.6 |
| 0 | 80.2 / 80.8 / 81.6 | 33.7 / 33.4 / 31.7 |
| -6 | 78.2 / 79.9 / 78.8 | 37.2 / 35.3 / 38.4 |

and *drawer* sound events was very low, making them difficult to detect when there was background noise, resulting in many deletion errors. Even humans have difficulty detecting these sounds under low SNR conditions. The poor results for *pageturn* occurred for a different reason, however. One reason for the low detection performance for *pageturn* was a large number of insertion errors due to the difference in the background noise between the training and evaluation sets. The background noise in the development set is almost the same as in the training set, and in that case we did not observe a large number of insertion errors. However, background noise in the evaluation set is different from that in the training set, so there is a possibility that the network focused on a specific pattern in the feature vector, similar to one observed in the background noise of the evaluation set.

Focusing now on the differences in performance among BLSTM-based methods, we can see that the overall trends for each model were similar. However, the combination of the BLSTM with an HMM or HSMM was not always effective for all of the sound event classes. To investigate this in detail, performance under different SNR conditions was examined and the results are shown in Table 4.3, which again displays the performance after both post-processing and SAD binary masking have been applied. From these results we can see that

Table 4.4: *Details of segment-based error rates without SAD masking.*

| | Segment-based Error rate (BLSTM / HMM / HSMM) | | |
|---|---|---|---|
| SNR [dB] | S [%] | D [%] | I [%] |
| 6 | 8.2 / 8.0 / 7.1 | 4.6 / 5.4 / 3.6 | 22.2 / 22.1 / 20.7 |
| 0 | 8.9 / 7.6 / 7.6 | 4.9 / 5.2 / 4.5 | 22.3 / 23.1 / 23.1 |
| -6 | 9.0 / 7.8 / 8.2 | 7.6 / 6.6 / 5.3 | 22.7 / 24.4 / 30.9 |

the performance of all of the models degraded under the low SNR condition, and that the BLSTM-HSMM method was especially susceptible to the effect of background noise. The details of the segment-based error rates without SAD masking are shown in Table 4.4, and those with SAD masking are shown in Table 4.5, where S, D, and I represent the substitution, deletion, and insertion error rates, respectively. Focusing on the substitution error rate, we can confirm that the combination of the BLSTM with an HMM or HSMM was always effective for the reduction of substitution errors. This is because each sound event has a different duration, and therefore substitution errors can be reduced by modeling the duration of each sound event. However, regarding the insertion error rate, while the BLSTM and BLSTM-HMM maintained their performance with only a slight degradation under the low SNR condition, the performance of BLSTM-HSMM fell drastically. This is because the BLSTM-HSMM method models the duration of sound events, forcing them in particular to sustain a certain length, and the mistakenly inserted sound events were thus of significant duration. This caused long-term frame mistakes, and consequently, the performance decreased drastically. The use of SAD network binary masking improved performance by reducing the number of insertion errors, however, there was still a gap between the performance of the

Table 4.5: *Details of segment-based error rates with SAD masking.*

| | Segment-based Error rate (BLSTM / HMM / HSMM) | | |
|---|---|---|---|
| SNR [dB] | S [%] | D [%] | I [%] |
| 6 | 8.2 / 8.2 / 7.3 | 4.8 / 5.9 / 5.5 | 19.1 / 18.7 / 15.9 |
| 0 | 8.7 / 7.6 / 7.4 | 5.4 / 6.0 / 6.0 | 19.5 / 19.8 / 18.4 |
| -6 | 9.0 / 7.6 / 7.8 | 7.9 / 7.1 / 6.2 | 20.3 / 20.6 / 24.5 |

Table 4.6: *Performance of system combination.*

| Criteria | F1 score [%] | Error rate [%] |
|---|---|---|
| Event-based | 76.3 (+1.0) | 42.0 (-2.2) |
| Segment-based | 82.4 (+1.3) | 30.2 (-2.7) |

BLSTM-HSMM and the other methods under the low SNR condition. As a result, although the BLSTM-HSMM method achieved the best results on average, by examining details of the performance of each method we can see that each model has advantages and disadvantages. Hence, we hypothesized that it would be advantageous to develop a model which combines the feature of each of the BLSTM-based methods. To confirm this, a system combination was devised in which the majority result of the outputs of the three methods was selected as the output, after which post-processing and SAD masking were applied. Performance results for the system combination are shown in Table 4.6, where the numbers in parentheses represent the amount of improvement from the best results in Table 4.2. The combination of

Table 4.7: *Performance for different tasks.*

| | Segment-based (BLSTM / HMM / HSMM) | |
|---|---|---|
| Task | F1 score [%] | Error rate [%] |
| polyphonic | 79.2 / 79.5 / 80.0 | 34.7 / 34.0 / 33.0 |
| monophonic | 81.1 / 82.4 / 82.9 | 33.7 / 33.4 / 32.8 |

the three methods achieved the best performance, which supports the above hypothesis.

Finally, we focus on differences in performance when performing monophonic versus polyphonic SED task. The results are shown in Table 4.7. Somewhat surprisingly, all of the models achieved similar performance on both tasks, and there was only a slight difference in performance between the monophonic task and the much more challenging polyphonic task.

## 4.6 Summary

In this chapter, a new hybrid approach for polyphonic SED called duration-controlled LSTM was proposed, which incorporates a duration-controlled modeling technique based on an HSMM and a frame-by-frame detection method based on a BLSTM. The proposed duration-controlled LSTM made it possible to model the duration of each sound event explicitly and to perform sequence-by-sequence detection without the need for thresholding. Furthermore, to reduce insertion errors which often occur under noisy conditions, a post-processing step using an SAD network was introduced to identify segments with any kind of sound event activity. The proposed method outperformed not only conventional methods such as NMF and standard BLSTM, but also the best results submitted for the DCASE2016 task2 Challenge, proving that the modeling of sound event duration is effective for polyphonic SED. Furthermore, combining the three BLSTM-based methods allowed further improvements.

# 5 Anomalous Sound Event Detection based on Waveform Modeling

This chapter describes a new method of anomalous sound event detection for use in public spaces. The proposed method utilizes WaveNet, a generative model based on an autoregressive convolutional neural network (CNN), to model various acoustic patterns in the time domain. When the model detects unknown acoustic patterns, they are identified as anomalous sound events. WaveNet has been used in generation tasks, such as speech synthesis, to precisely model a waveform signal and then directly generate it using random sampling. In contrast, the proposed method uses WaveNet as a predictor rather than as a generator to detect waveform segments that causes large prediction errors as anomalous acoustic patterns. Because WaveNet is capable of modeling detailed temporal structures like the phase of waveform signals, the proposed method is expected to detect anomalous sound events more accurately than conventional methods based on the reconstruction errors of acoustic features. Furthermore, to take differences in environmental situations into consideration, i-vector is used as an additional auxiliary feature of WaveNet, which has been utilized in the field of speaker verification. This i-vector extractor will enable discrimination of the sound patterns, depending on the time, location, and the surrounding environment. Experimental evaluation using a database recorded in public spaces demonstrate that the proposed method outperforms conventional feature-based approaches, and that modeling in the time domain in conjunction with the i-vector extractor is effective for anomalous SED.

## 5.1 Introduction

In response to crime and the rising number of terrorism incidents, demands for better public safety have been increasing all over the world. To meet these demands, video-based monitoring systems have been developed which make it possible to automatically detect suspicious people or objects [134, 135], as well as sound-based security systems which can automatically detect anomalous sound events such as glass breaking [13–15]. Video-based systems have proven to be effective, however, due to blind spots and physical, social, political and economic limits on the installation of cameras, it is difficult for these systems to monitor an entire area. On the other hand, sound-based systems have been attracting increased attention because they have no blind spots, and microphones cheaper are easier to install than cameras. Therefore, sound-based systems can complement video-based systems by covering camera blind spots and areas where cameras would be inappropriate, such as public restrooms or changing rooms. Furthermore, a combination of sound-based and video-based systems is likely to result in more intelligent monitoring systems.

The key technology behind sound-based monitoring system can be divided into two categories; supervised and unsupervised approaches. Supervised approaches use manually labeled data, and employ acoustic scene classification (ASC) [37,43] and acoustic event detection (AED) [51, 55, 136] methods. Here, scenes represent the environment which the audio segments are recorded, and sound events represent sounds related to human behaviors or the motion or contact of objects. The task of ASC is to classify long-term audio segments into pre-defined scenes, while the task of AED is to identify the start and end times of pre-defined sound events in order to label them. These technologies make it possible to understand an environment and detect various types of sounds, but they require the pre-definition of all of the possible scenes and sound events, and the collection and labeling of large amounts of this type of sound data are difficult.

Unsupervised approaches, on the other hand, do not require manually labeled data, so they are less costly to develop. One unsupervised approach is change-point detection [76–78], which compares a model of the current time with that of a previous time to calculate a dissimilarity score, and then identifies highly dissimilar comparison results as anomalies. However, in the case of public spaces, the sounds which can occur are highly variable and non-stationary, and therefore, the detected change points are not always related to anomalies that are of concern (e.g., the sound of the departure of the train). Another unsupervised approach is outlier detection [79–81], which models an environment's normal sound patterns, and then detects patterns which do not correspond to the normal model identifying them as anomalies. Note that the normal patterns are patterns which have appeared in the training data. Typically, a Gaussian mixture model (GMM) or one-class support vector machine (SVM) with acoustic features such as Mel-frequency cepstrum coefficients (MFCCs) is used [82, 83]. Thanks to recent advances in deep learning, neural network-based methods are now attracting attention [84–86]. These methods train an auto-encoder (AE) or a long short-term memory recurrent neural network (LSTM-RNN) with only normal scene data. While an AE encodes the inputs as latent features and then decodes them as the original inputs, an LSTM-RNN predicts the next input from the previous input sequence. Using a trained model, reconstruction errors between observations and the predictions are calculated, and high error patterns are identified as anomalies. Although these methods have achieved good performance, it is difficult to directly model acoustic patterns in the time domain due to their highly non-stationary nature and their high sampling rates so they typically use feature vectors extracted from audio signals.

In this chapter, a new method of detecting anomalous sound events in public spaces is proposed, which utilizes WaveNet [16, 137, 138], a generative model based on an autoregressive convolutional neural network, to directly model the various acoustic patterns occurring in public spaces in the time domain. Based on this model, unknown acoustic patterns are identified as anomalous sound events. WaveNet has been used in generation tasks such as speech synthesis to precisely model a waveform signal and then directly generate it using random sampling. The proposed method uses WaveNet as a predictor rather than as a generator, however, to detect waveform segments that cause large prediction errors as anomalous acoustic patterns. Because WaveNet is capable of modeling the detailed temporal structures like the phase of waveform signals, the proposed method is expected to detect anomalous sound events more accurately than conventional methods based on the reconstruction errors of acoustic features. Furthermore, to take differences in environmental situations into consideration, i-vector is used as an additional auxiliary feature of WaveNet, which has been utilized in the field of speaker verification. This i-vector extractor will enable discrimination of the sound patterns, depending on the time, location, and the surrounding environment. Experimental evaluation using a database of sounds recorded in public spaces demonstrate that the proposed method can outperform conventional feature-based anomalous SED approaches, and that modeling in the time domain in conjunction with the use of i-vector is effective for anomalous SED.

The rest of this chapter is organized as follows: Section 5.2 provides an overview of WaveNet. Section 5.3 explains the basics of i-vector approach. Section 5.4 describes the proposed anomalous SED system. Section 5.5 discusses the design of the experiment and evaluates the performance of the proposed method. Finally, this chapter is summarized in Section 5.6.

## 5.2 Overview of WaveNet

### 5.2.1 WaveNet

WaveNet is an autoregressive CNN which is capable of directly generating raw audio waveforms [16]. The joint probability of a waveform $\mathbf{x} = \{x_1, x_2, \ldots, x_{n-1}\}$ is factorized as a product of conditional probabilities as follows:

$$p(\mathbf{x}) = \prod_{n=1}^{N} p(x_n | x_1, x_2, \ldots, x_{n-1}). \tag{5.1}$$

WaveNet approximates the conditional probability above by canceling the effect of past samples of a finite length as follows:

$$\text{WaveNet}(\mathbf{x}) \simeq p(x_n | x_{n-R-1}, x_{n-R}, \ldots, x_{n-1}, \mathbf{\Lambda}), \tag{5.2}$$

where $\mathbf{\Lambda}$ is set of trainable network parameters, and $R(> 0)$ is the number of past samples to be taken into account, which is known as the receptive field. In order to model a waveform directly, it is necessary to secure very large receptive fields since a waveform consists of tens of thousands of samples. As a result, direct modeling involves huge computational costs. To address this issue, WaveNet utilizes dilated causal convolution, the use of convolutions with holes where the output does not depend on future samples. A visualization of a stack of dilated causal convolution layers is shown in Fig. 5.1. This architecture can secure very large receptive fields while also significantly reducing computational costs. Another important feature of WaveNet is its use of quantized waveforms, which are generally quantized to 256 values using $\mu$-law algorithm [139]. By using quantized waveforms, WaveNet can estimate the posteriors of discrete amplitude classes instead of their continuous values. Therefore, WaveNet is optimized using a classification task rather than a regression task, making the training process much easier to optimize.

An overview of the structure of WaveNet is shown in Fig. 5.2(a). WaveNet consists of

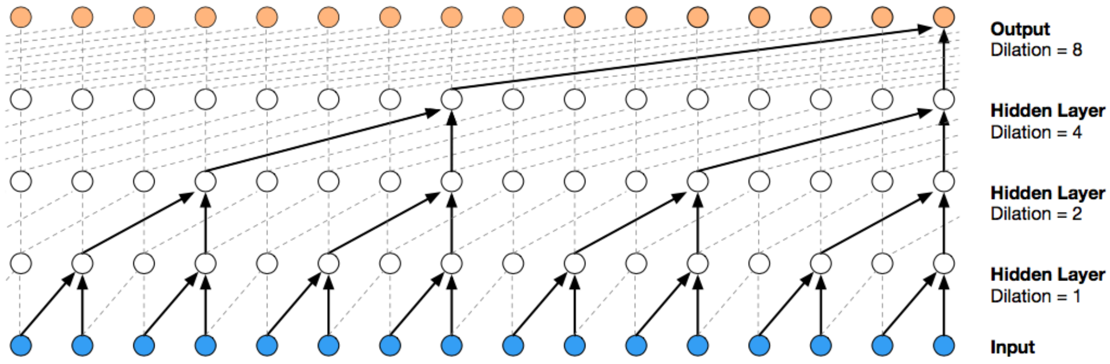


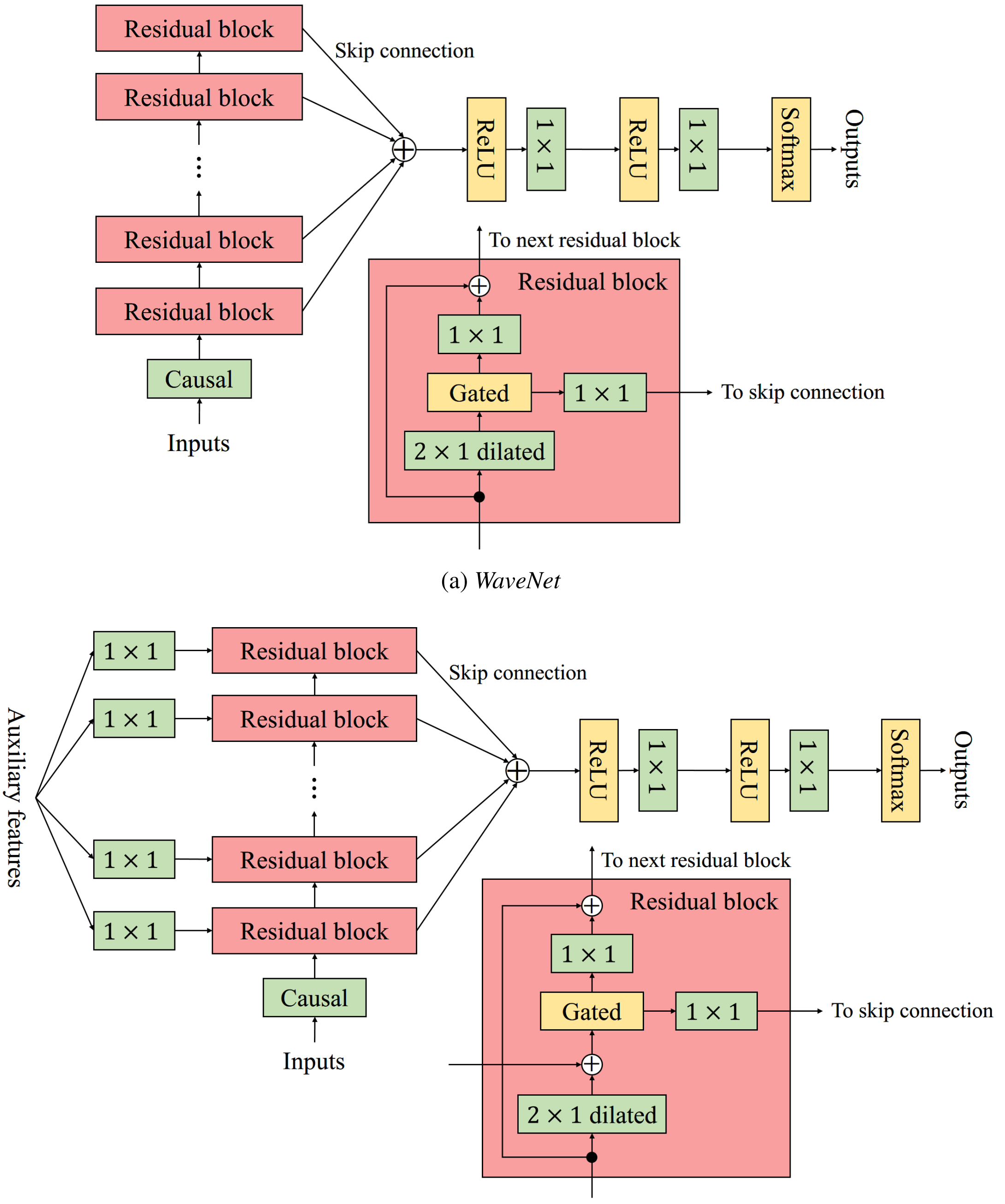
Figure 5.1: *Visualization of a stack of dilated causal convolution layers [16].*

many residual blocks, each residual block consists of 2 × 1 dilated causal convolutions, a gated activation function, and 1 × 1 convolutions. Each residual block is connected to the final output layers with a skip-connection. This residual structure makes it possible to train a much deeper network, preventing the gradient vanishing problem [41, 124]. The gated activation function is calculated as follows:

$$\mathbf{z} = \tanh\left(\mathbf{W}_{f,k} * \mathbf{x}\right) \odot \sigma\left(\mathbf{W}_{g,k} * \mathbf{x}\right), \tag{5.3}$$

where $\mathbf{W}$ is a trainable convolution filter, $\mathbf{W} * \mathbf{x}$ represent a dilated causal convolution, $\odot$ represents element-wise multiplication, $\sigma(\cdot)$ represents a sigmoid activation function, $k$ is the index of a residual block, $f$ and $g$ represent filter and gate, respectively. The term of the sigmoid activation function works as a gate to determine whether to pass along the outputs of the filter. The dilation is doubled for every layer up to a certain point and then repeated (e.g. $1, 2, 4, \ldots, 512, 1, 2, 4, \ldots, 512$).

During training, WaveNet is used as a finite impulse response (FIR) filter, i.e., it predicts a future sample $x_t$ from observed samples $x_{t-R-1:t-1}$. This technique is also known as teacher forcing learning. WaveNet is optimized through back-propagation using the following cross-



(a) *WaveNet*

(b) *Conditional WaveNet*

Figure 5.2: *Overview of WaveNet architectures. "Causal", "1 × 1", and "dilated" represent causal, 1 × 1, and dilated causal convolution, respectively, "ReLU", "Gated", and "Softmax" represent rectifier linear unit, gated and Softmax activation functions, respectively.*

entropy objective function:

$$E(\mathbf{\Lambda}) = -\sum_{n=1}^{N}\sum_{q=1}^{Q} y_{t,q} \log \hat{y}_{t,q} \tag{5.4}$$

where $\mathbf{y}_n = \{y_{n,1}, y_{n,2}, \ldots, y_{n,Q}\}$ represents the one-hot vector of the target quantized waveform signal, $\hat{\mathbf{y}}_n = \{\hat{y}_{n,1}, \hat{y}_{n,2}, \ldots, \hat{y}_{n,Q}\}$ represents the posterior of the amplitude class, $t$ and $i$ represent the index of the waveform samples and their amplitude class, respectively, and $Q$ represents the number of quantized values.

On the other hand, during decoding WaveNet is used as an autoregressive filter, i.e., it predicts future sample $\hat{x}_t$ from predicted samples $\hat{x}_{t-R-1:t-1}$. WaveNet repeats this procedure until reaching the target length to generate a waveform, however, it requires a huge amount of time to generate a long waveform. To address this issue, a fast decoding process and parallel WaveNet have been proposed [140, 141].

### 5.2.2 Conditional WaveNet

By using additional auxiliary features $\mathbf{h}$, WaveNet can model conditional probability $p(\mathbf{x}|\mathbf{h})$. Equation (5.1) can then be transformed into the following equation:

$$p(\mathbf{x}|\mathbf{h}) = \prod_{n=1}^{N} p(x_n|x_1, x_2, \ldots, x_{n-1}, \mathbf{h}). \tag{5.5}$$

By conditioning WaveNet with auxiliary features, we can control the characteristics of the generated samples. In original WaveNet [16], linguistic features and/or speaker codes are conditioned to generate speech samples based on given text information while keeping specific speaker characteristics. In other studies [137, 138, 142], acoustic features such as Mel cepstrum, $F_0$, or log Mel-spectrogram are used as auxiliary features to make WaveNet function as a vocoder.

The architecture of conditional WaveNet is shown in Fig. 5.2(b), and Equation (5.3) is replaced as follows:

$$\mathbf{z} = \tanh\left(\mathbf{W}_{f,k} * \mathbf{x} + \mathbf{V}_{f,k} * f(\mathbf{h})\right) \odot \sigma\left(\mathbf{W}_{g,k} * \mathbf{x} + \mathbf{V}_{g,k} * f(\mathbf{h})\right), \tag{5.6}$$

where $\mathbf{V}$ is a trainable convolution filter, $\mathbf{V} * f(\mathbf{y})$ represents $1 \times 1$ convolution, and $f(\cdot)$ is a function which makes the length of the auxiliary features the same as that of the input waveform.

## 5.3 Overview of i-vector

### 5.3.1 GMM-UBM

One of the generative approaches in the field of speaker verification is Gaussian mixture model - universal background model (GMM-UBM) [143]. A GMM-UBM, which has a relatively large number of mixture components (e.g. 2,048), is trained using a large amount of multiple speakers' data and then maximum a prosteriori (MAP) [143] or maximum likelihood linear regression (MLLR) [144] adaptation is performed using a limited amount of a specific speaker data in order to build a speaker-dependent model. Finally, a speaker similarity score is calculated based on the ratio of the likelihood of the speaker-dependent model and that of UBM to perform an identification task.

As an extension of the GMM-UBM approach, a GMM supervector has also been proposed [145]. GMM supervector $\mathbf{M}$ for a given audio segment $\mathbf{u} = \{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_L\}$ is represented as follows:

$$\mathbf{M}_u = \left[\mathbf{m}_{u,1}^\top, \mathbf{m}_{u,2}^\top, \ldots, \mathbf{m}_{u,C}^\top\right]^\top, \tag{5.7}$$

where $C$ is the number of mixtures, and $\mathbf{m}_{u,c}$ represents a mean vector of the $c$-th mixture of the adapted GMM using a given audio segment $\mathbf{u}$. GMM supervectors have generally been used as the input for discriminative models such as SVM [145], achieving better performance compared to the GMM-UBM approach.

### 5.3.2 i-vector

i-vector is a fixed, low-dimensional representation of audio segments which contains information about the difference between an adapted distribution and a universal background model (UBM) in a total variability space defined by factor analysis [33]. A visualization of the concept of i-vector is shown in Fig. 5.3, where i-vector is represented by the low dimensional representation of the two arrows, which represent the difference between the adapted distribution and the UBM.

Using factor analysis, the GMM supervector for a given audio segment **u** can be written as follows:

$$\mathbf{M}_u = \mathbf{m} + \mathbf{T}\mathbf{w}_u, \tag{5.8}$$

where **m** represents the UBM supervector, **T** is a rectangular matrix called a total variability matrix, and **w** is an i-vector. First, in order to calculate the i-vector for a given audio segment **u**, first and second order Baum-Welch statistics ($N_{u,c}$ and $\mathbf{F}_{u,c}$) are calculated as follows:

$$N_{u,c} = \sum_{t=1}^{L} p(c|\mathbf{u}_t) \tag{5.9}$$

$$\mathbf{F}_{u,c} = \sum_{t=1}^{L} p(c|\mathbf{u}_t)(\mathbf{u}_t - \mathbf{m}_c), \tag{5.10}$$

where $\mathbf{m}_c$ is the mean of the $c$-th mixture of the UBM, and $p(c|\mathbf{u}_t)$ is a posterior of the mixture components of the UBM, which can be calculated as follows:

$$p(c|\mathbf{u}_t) = \frac{\pi_c p(\mathbf{u}_t|\mathbf{m}_c, \mathbf{\Sigma}_c)}{\sum_{k=1}^{C} \pi_k p(\mathbf{u}_t|\mathbf{m}_k, \mathbf{\Sigma}_k)}. \tag{5.11}$$

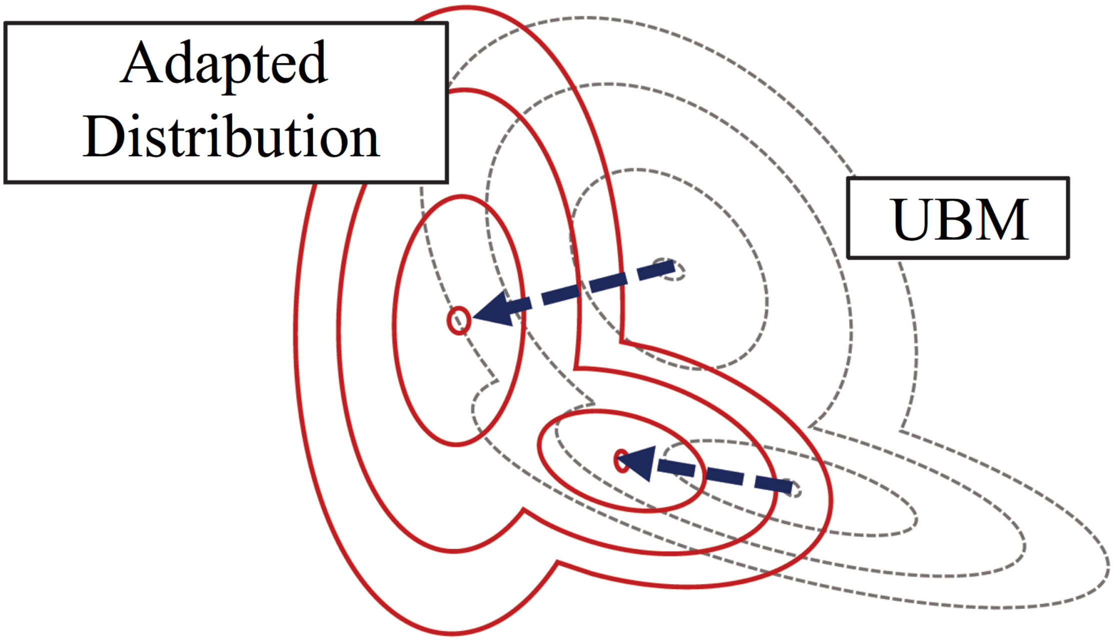


Figure 5.3: *Visualization of i-vector. i-vector is low dimensional representation of two arrows which represent the difference between the UBM and a distribution of each given audio segment.*

Then let us define $\mathbf{N}_u$ and $\mathbf{F}_u$ as follows:

$$
\begin{aligned}
\mathbf{N}_u &= \begin{bmatrix} N_{u,1}\mathbf{I}_D & 0 & \cdots & 0 \\ 0 & N_{u,2}\mathbf{I}_D & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & N_{u,C}\mathbf{I}_D \end{bmatrix}, \qquad (5.12) \\
\mathbf{F}_u &= \left[\mathbf{F}_{u,1}^{\top}, \mathbf{F}_{u,2}^{\top}, \ldots, \mathbf{F}_{u,C}^{\top}\right]^{\top}, \qquad (5.13)
\end{aligned}
$$

where $\mathbf{I}_D$ is a $D \times D$ identity matrix, and $D$ is the dimension of the representation of an audio segment. Finally, using these terms, i-vector is calculated as follows:

$$\mathbf{w}_u = \left(\mathbf{I} + \mathbf{T}^{\top}\mathbf{\Sigma}^{-1}\mathbf{N}_u\mathbf{T}\right)^{-1} \mathbf{T}^{\top}\mathbf{\Sigma}^{-1}\mathbf{F}_u. \qquad (5.14)$$

Since the i-vector contains information about not only the speaker and but also the channel (e.g. recording condition), it is necessary to reduce the channel information in order to improve speaker recognition or verification performance. To address this issue, various dimension reduction methods have been proposed, such as within-class covariance normalization (WCCN) [146] and probabilistic linear discriminant analysis (PLDA) [147, 148].

## 5.4 Proposed Method

### 5.4.1 System overview

An overview of the proposed anomalous SED system, separated into training and test phases, is shown in Fig. 5.4. In the training phase, audio segments of 30 seconds each are divided into 25 ms windows to calculate two acoustic features; 40-dimensional log Mel filter banks with 96 % overlap and 13-dimensional MFCCs + Δ + ΔΔ with 40 % overlap. A GMM-UBM with 256 mixture components is trained using the MFCCs of each segment, then total variability matrix and covariance matrix are estimated in order to extract a 60-dimensional

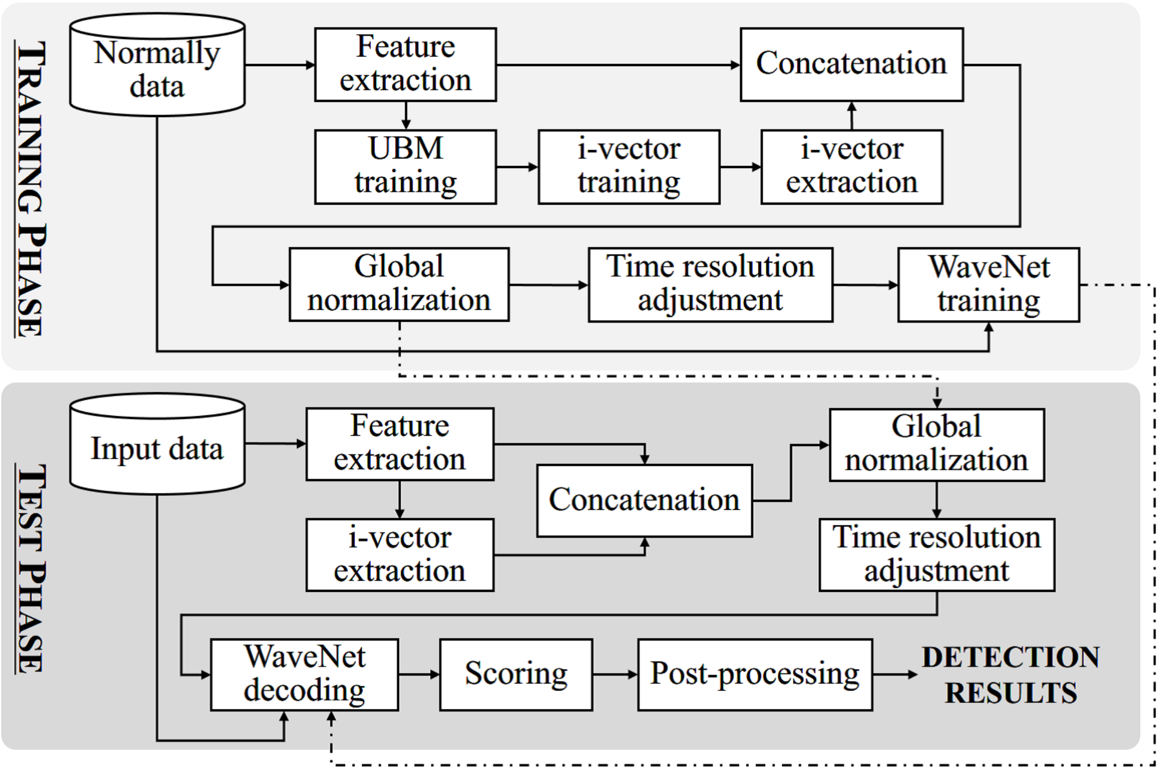


Figure 5.4: *Overview of the proposed anomalous SED system.*

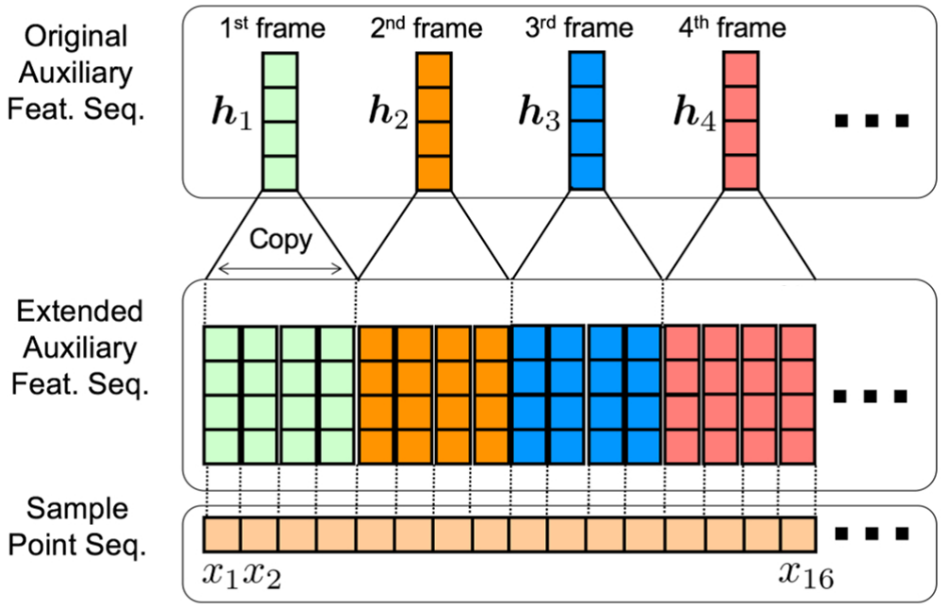
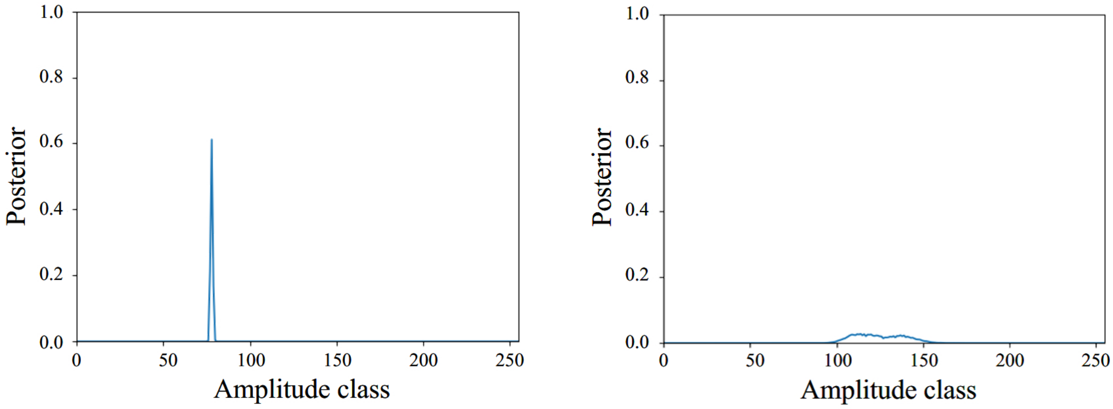


Figure 5.5: *Time resolution adjustment procedure.*

i-vector. Using the estimated matrices, i-vector is extracted for each audio segment, then log Mel filter banks are concatenated with a replicated i-vector, and used as auxiliary features for WaveNet. Statistics of the auxiliary features are then calculated using the training data in order to perform global normalization, making the mean and variance of each dimension of the features 0 and 1, respectively. The time resolution adjustment procedure (shown in Fig. 5.5) is then performed to make the time resolution of the features the same as the waveform signal. The input waveform is quantized into 8 bits using the $\mu$-law algorithm [139] and then converted into a sequence of 256-dimensional one-hot vectors. Finally, WaveNet is trained using the one-hot sequences of quantized waveforms and the corresponding auxiliary features.

In the test phase, as in the training phase, auxiliary features are extracted from the input audio segment and normalized using the statistics of the training data. The input waveform signal is also quantized and then converted into a sequence of one-hot vectors. WaveNet then



(a) *Posterior of a known sound*

(b) *Posterior of a unknown sound*

Figure 5.6: *Examples of WaveNet posteriors for known and unknown sounds.*

calculates a posteriogram (a sequence of posteriors) with sequence of one-hot vectors of the quantized waveform and the auxiliary features. Note that in the proposed method WaveNet is used as a finite impulse response (FIR) filter in decoding, therefore, it operates much faster than an autoregressive filter. Next, the entropy of each posterior is calculated over the posteriogram and thresholding is performed for the sequence of entropies in order to detect anomalies Finally, three kinds of post-processing are performed to smooth the detection result.

The scoring and post-processing procedures are described in detail in Sections 5.4.2 and 5.4.3, respectively.

### 5.4.2 Scoring

To detect anomalous sound events, the uncertainty of the WaveNet prediction is focused. Examples of the posteriors of WaveNet for known and unknown sounds are shown in Fig. 5.6. As we can see, the shape of the posterior of a known sound is sharp while that of an unknown sound is flat. Hence, it is expected that we can identify unknown sounds as anomalous sound events based on the uncertainty of the prediction.

To quantify the uncertainty of the prediction, entropy $e$ of the posterior is calculated as follows:

$$e_t = -\sum_{c=1}^{C} \hat{y}_{t,c} \log_2 \hat{y}_{t,c}. \tag{5.15}$$

The entropy is calculated over the posteriogram, resulting in the entropy sequence $\mathbf{e} = \{e_1, e_2, \ldots, e_T\}$. Finally, thresholding for the sequence of entropies is performed using the following threshold value:

$$\theta = \mu + \beta\sigma, \tag{5.16}$$

where $\theta$ represents the threshold value, $\mu$ and $\sigma$ represent the mean and the standard deviation of the entropy sequence, respectively, and $\beta$ is a hyperparameter. The value of parameter $\beta$ is decided through preliminary experiments.

An example of a sequence of entropies is shown in Fig. 5.7. We can see that entropy increases in the section of the sequence corresponding to the unknown sound.

### 5.4.3 Post-processing

To smooth the detection results, three kinds of post-processing are used:

1. Apply a median filter with a predetermined filter span;
2. Fill gaps which are shorter than a predetermined length;
3. Remove events whose duration is shorter than a predetermined length.

Diagrams of each post-processing step are shown in Fig. 5.8. The parameters for post-processing are decided through preliminary experiments.

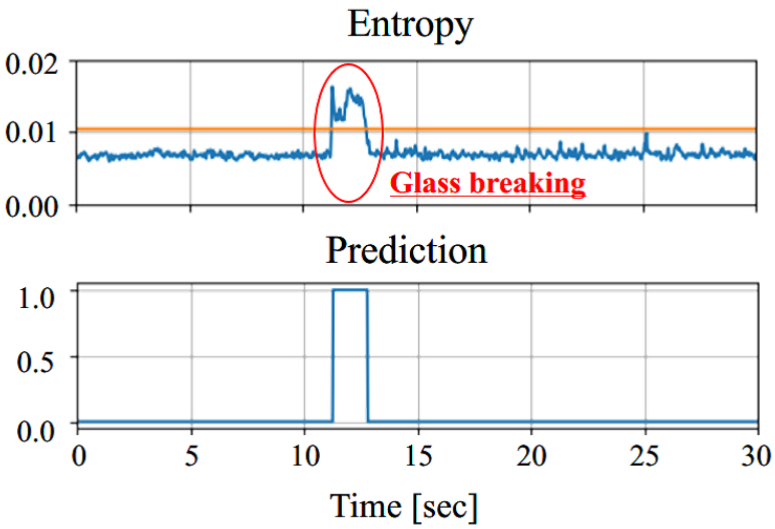
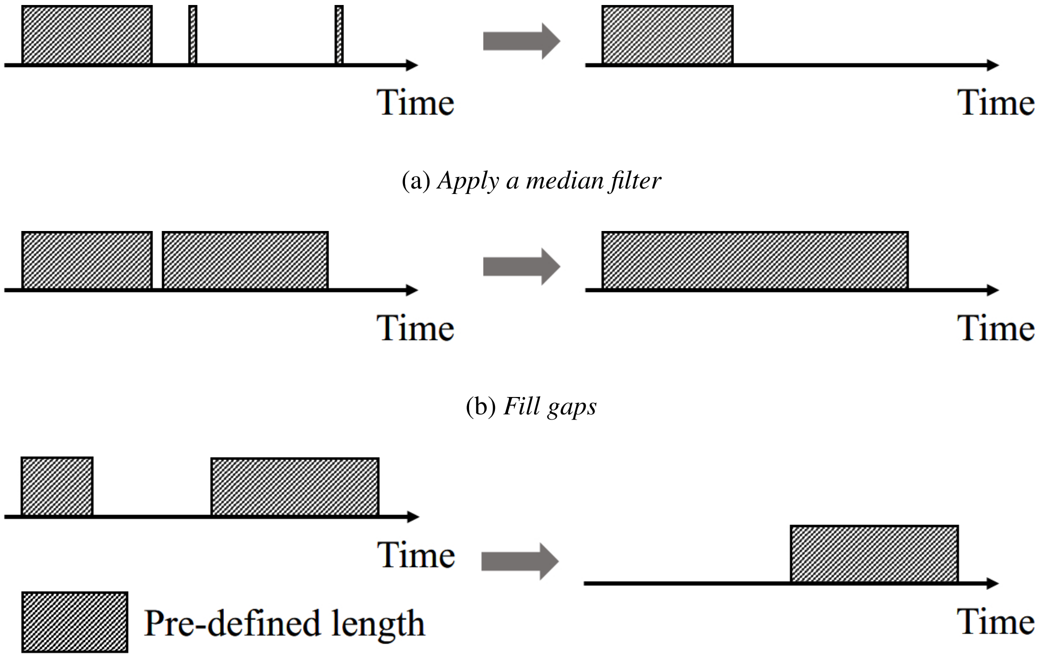


Figure 5.7: *Example of a sequence of entropies and their threshold values (top) and the corresponding binarized detection results (bottom).*

## 5.5 Experimental Evaluation

### 5.5.1 Experimental conditions

The proposed anomalous SED method was evaluated using two-weeks of audio data recorded at a subway station. Data from the first week was used as training data, and the rest of the data was used as evaluation data. The continuous audio data was divided into 30-second segments and anomalous sounds were added to each piece of evaluation data. The added anomalous sounds included the sound of glass breaking, various types of human screams, and human growling, which were selected from the Sound Ideas Series 6000 General Sound Effects Library [149]. Each sound was added at random temporal positions with three signal-to-noise ratios (SNRs): 0 dB, 10 dB, and 20 dB. Evaluations were conducted under two regimes; using an event-based metric (onset only), and using a segment-based



(a) *Apply a median filter*

(b) *Fill gaps*

(c) *Remove short duration events*

Figure 5.8: *Diagram of each post-processing step.*

evaluation metric, where the F1-score was used as the evaluation criteria [132].

The performance of the proposed method was compared to the following methods:

- Auto-encoder (AE),
- Auto-regressive LSTM (AR-LSTM),
- Bidirectional LSTM auto-encoder (BLSTM-AE),
- WaveNet without i-vector (WAVENET W/O I-VEC).

Each of top three networks consisted of 3 hidden layers with 256 hidden units, and the inputs were 40-dimensional log Mel filter bank features, which were extracted with a 25 ms window and a 10 ms shift. All of these networks were optimized using Adam [106] under the objective function based on mean squared error. Thresholding and post-processing were the same as the proposed method. All of the networks were trained using the open source toolkit Keras [150] and TensorFlow [133] with a single GPU (Nvidia GTX 1080Ti). The extraction of i-vectors was performed using Kaldi [151].

### 5.5.2 Experimental results

The experimental results are shown in Fig. 5.9, where EB and SB represent event-based and segment-based metrics, respectively. First, we will compare the performance of the proposed WaveNet-based methods with that of conventional, feature-based methods. Based on the results shown in Fig. 5.9, the proposed WaveNet-based methods outperformed conventional methods for both event-based and segment-based metrics, confirming its effectiveness. Thus, it is implied that modeling in the time domain, which captures the temporal structure of sound, is effective for anomalous SED.

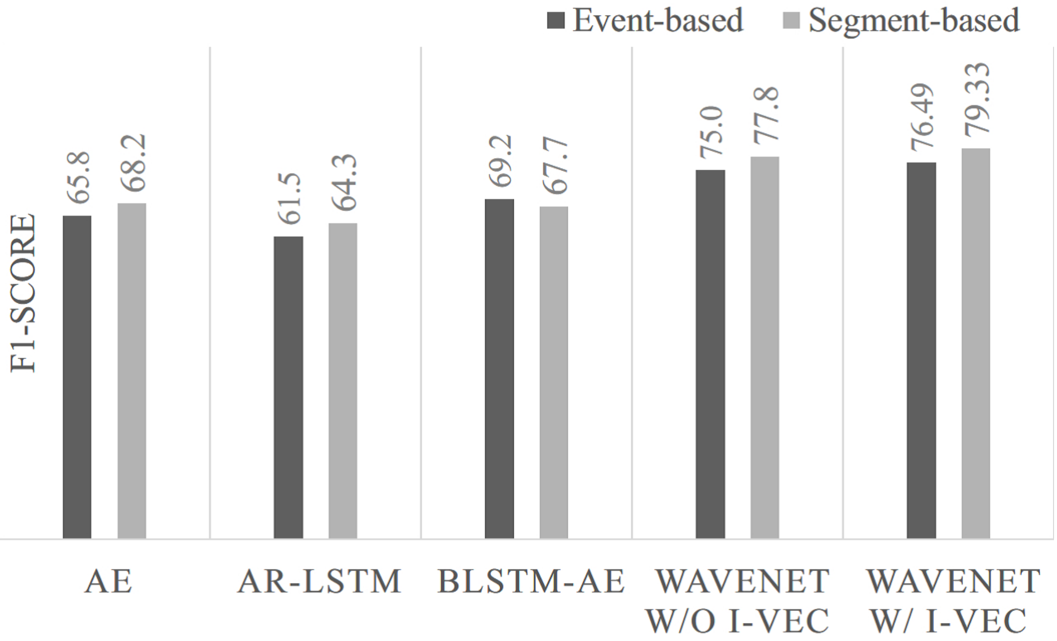


Figure 5.9: *Experimental results. EB and SB represent event-based and segment-based metrics, respectively.*

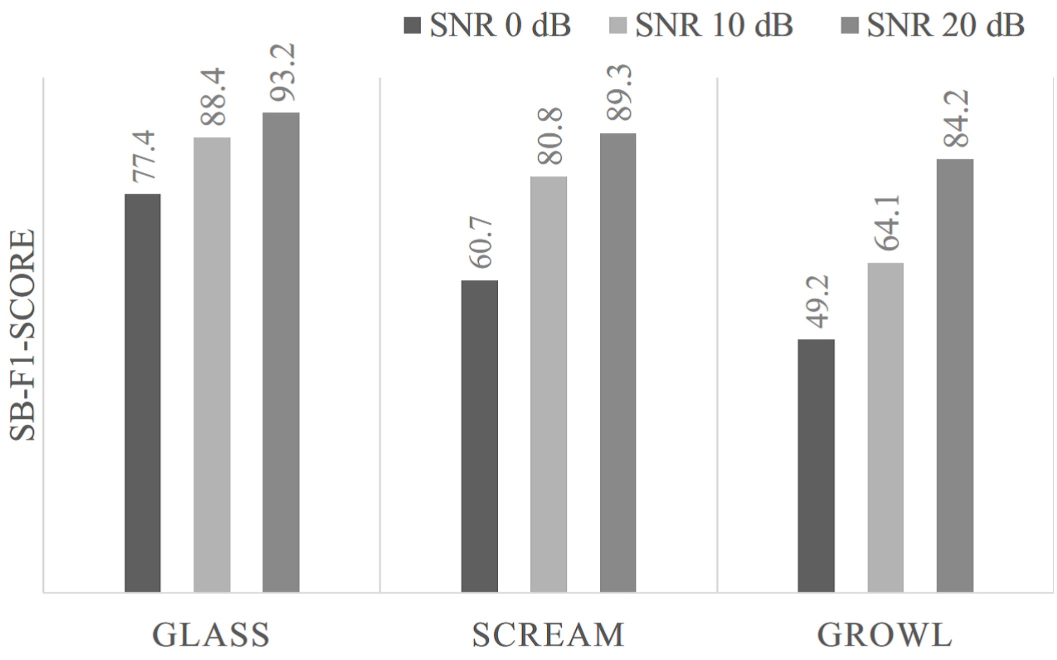
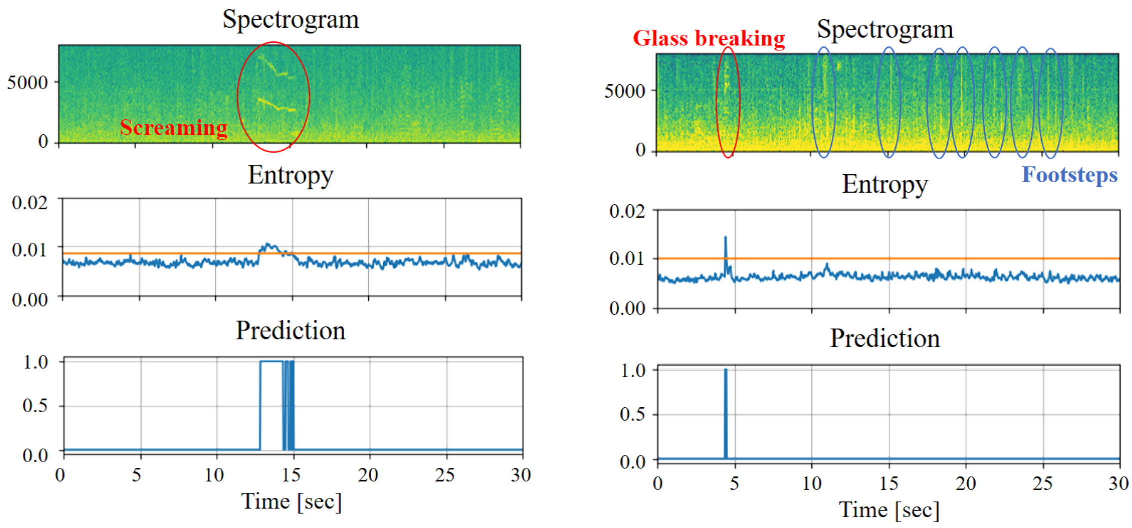


Figure 5.10: *Experimental results for each sound class and each SNR condition. SB represents segment-based metric. GLASS, SCREAM, and GROWL represent the sound of glass breaking, various types of human screaming, and human growling, respectively.*

Next, we compare performance between WaveNet with and without i-vector. The experimental results show that the proposed method (WaveNet with i-vector) outperforms the model without i-vector for both event-based and segment-based metrics. Thus, we can confirm that the use of i-vector, which represents environmental context, is effective for anomalous SED. It is also implied that the use of i-vector makes it possible to take into account differences in the meaning of sounds, depending on the environment in which they occur.

Finally, we focus on the difference in performance under each sound class and each SNR condition. The class-wise performance of the proposed method for each SNR condition is shown in Fig. 5.10. These results show that performance improved under higher SNR conditions, especially in the case of the sound of growling. This is because the power distribution



(a) *Detection of screaming*

(b) *Detection of glass breaking*

Figure 5.11: *Examples of detection results.*

in the frequency domain of growling largely overlapped with that of the chatting recorded in the training data. The reason why the proposed method achieved good performance for the sound of glass breaking, even under low SNR conditions is that, in contrast, it did not resemble any the sounds in the training data. The proposed method decides the hyperparameter $\beta$ globally in Equation (5.16), however, the results suggest that the best value of $\beta$ depends on types of anomalous sound events and environmental situations which are likely to be encountered. Therefore, if the anomalous SED method can dynamically adjust the hyperparameters for thresholding, it is likely that we could improve its performance, so this is an issue to be addressed in the future work.

Examples of the detection results are shown in Fig. 5.11. We can see that the proposed method can detect anomalous sound events even if they are difficult to distinguish in a spectrogram, and that it also can ignore similar patterns such as the footsteps of a woman wearing high heels.

## 5.6 Summary

In this chapter, a new anomalous SED method was proposed which utilizes WaveNet to directly model various acoustic patterns in the time domain. The proposed method uses WaveNet as a predictor, rather than as a generator, to detect waveform segments that cause large prediction errors as anomalous acoustic patterns. Because WaveNet is capable of modeling detailed temporal structures such as the phase information of the waveform signals, the proposed method can detect anomalous sound events more accurately than conventional methods which are based on the reconstruction errors of acoustic features. Furthermore, in order to take differences in environmental situations into consideration, i-vector was used as an additional auxiliary feature of WaveNet. This i-vector extractor allowed to discriminate sound patterns depending on the time, location, and surrounding environments. Experimental evaluation with a database recorded in public spaces showed that the proposed method outperformed conventional feature-based approaches, and that modeling in the time domain in conjunction with the i-vector extractor was effective for anomalous SED.

# 6 Relationship with Real World Data Circulation

The concept of real-world data circulation (RWDC) is the transfer of data from its acquisition to its analysis, and in turn, to its implementation, which is a key process for the success of a commercial application. This chapter introduces the concept of RWDC and discusses each work in terms of RWDC. This chapter also describes that how the developed tool for human activity recognition can be used to foster RWDC, how the analysis of data during research on polyphonic SED inspires a new method based on duration modeling, and how outputs from the research on anomalous SED can be used to improve the performance of polyphonic SED systems.

## 6.1 Introduction

For the success of a commercial application, real-world data must be collected and be utilized continuously. This process includes feedback from end users along with the analysis and application of that data for new products and services. This process of circulating data from acquisition, to analysis, and to implementation is known as real-world data circulation (RWDC) [17].

The concept of RWDC is shown in Fig. 6.1. In the acquisition phase, various phenomena in the real world are observed and this information is extracted as digital data. In the analysis phase, the acquired data is processed using information technologies such as machine

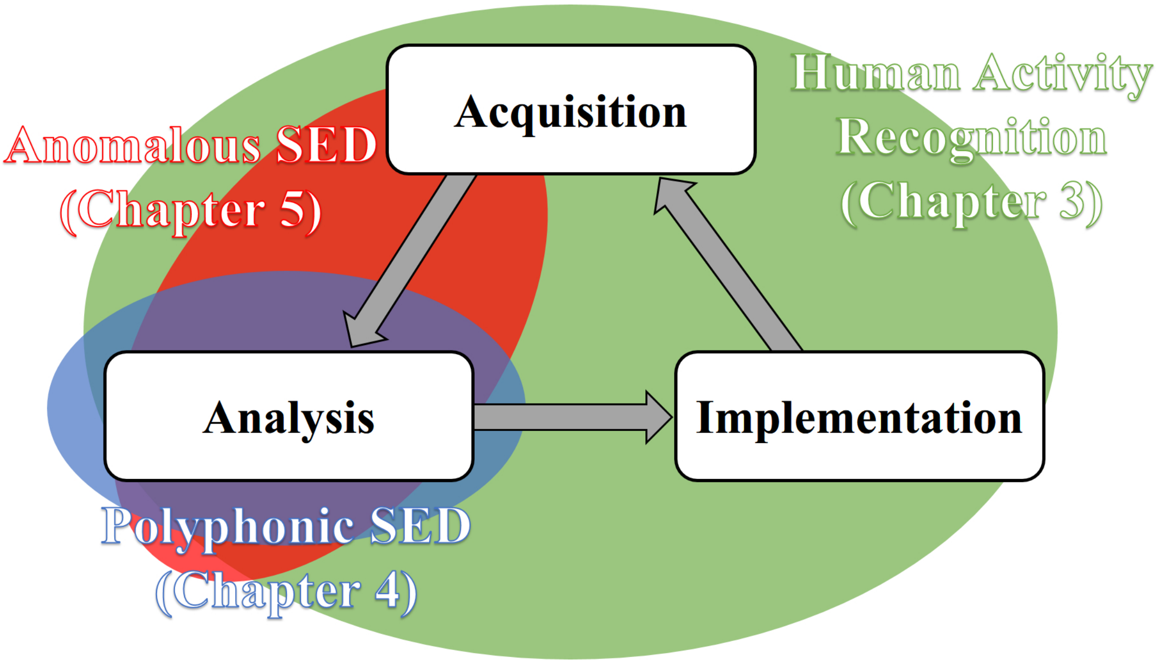

Acquisition
Analysis
Implementation
Anomalous SED (Chapter 5)
Polyphonic SED (Chapter 4)
Human Activity Recognition (Chapter 3)

Figure 6.1: *Concept of real-world data circulation. The acquisition phase represents the extraction of real-world digital data through the observation of various phenomena in the real world. The analysis phase represents the processing of the acquired data in order to create an overview or understanding of real-world phenomena using information technologies such as machine learning techniques. The implementation phase represents the incorporation of the knowledge acquired from the analyzed results into new services or products.*

learning techniques, and trends or associations are discovered which provide an overview of the phenomena being investigated. In the implementation phase, new services or products are created based on the results of analysis. By repeating this process of data circulation, we are able to feed the demands of end users back into the manufacturing process, resulting in the creation of a new social value.

In this chapter, each work in this thesis is discussed in terms of RWDC. Section 6.2 explains how the developed tools for human activity recognition can be used to foster RWDC. Section 6.3 describes how the analysis of data during research on polyphonic sound event detection (SED) inspires a new method for SED based on duration modeling. Section 6.4 explains how the output from the research on anomalous SED can be used to improve the performance of polyphonic SED systems, demonstrating the application of discovered knowledge to another process. Finally, this chapter is summarized in Section 6.5.

## 6.2 Human Activity Recognition based on DNN Using Multi-modal Signals

The work in the area of human activity recognition focused on the development of a method of realizing the process loop shown in Fig. 6.2. In this cycle, it is assumed that intellectual and physical activities result in experiences, such as the discovery of new knowledge or a sense of accomplishment. These experiences enhance people's abilities, expanding the range of activities open to them. Repeating this cycle continuously makes it possible for us to develop ourselves and improve our quality of life. In order to help people to keep repeating this cycle, it is necessary to monitor them and to understand their activities and experiences. To achieve this, a life-logging system was developed which can automatically record human activity, from simple movements such as walking to complex tasks such as

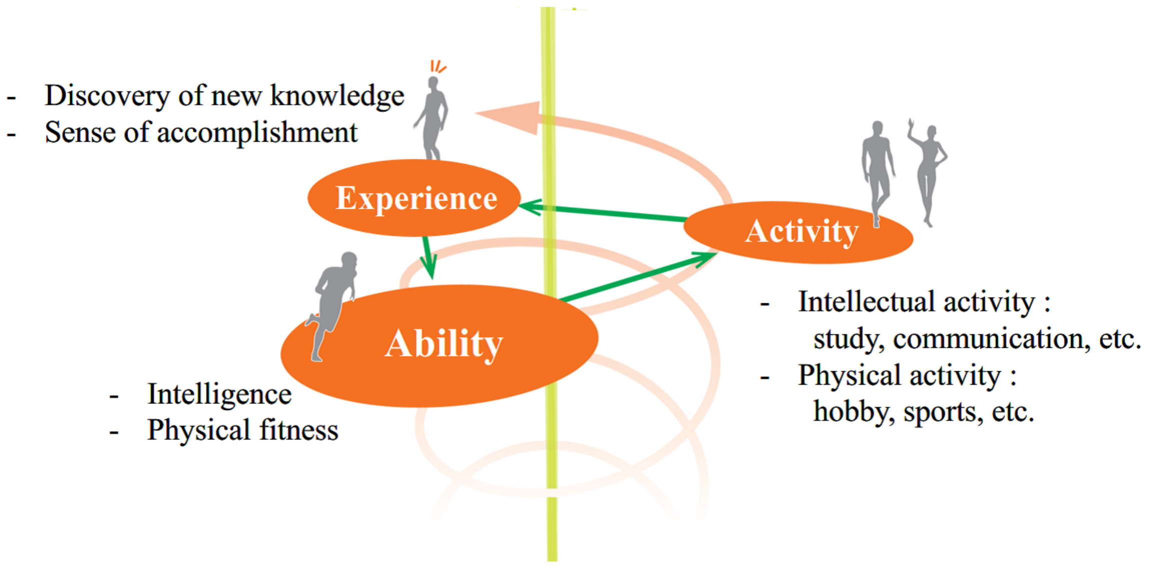
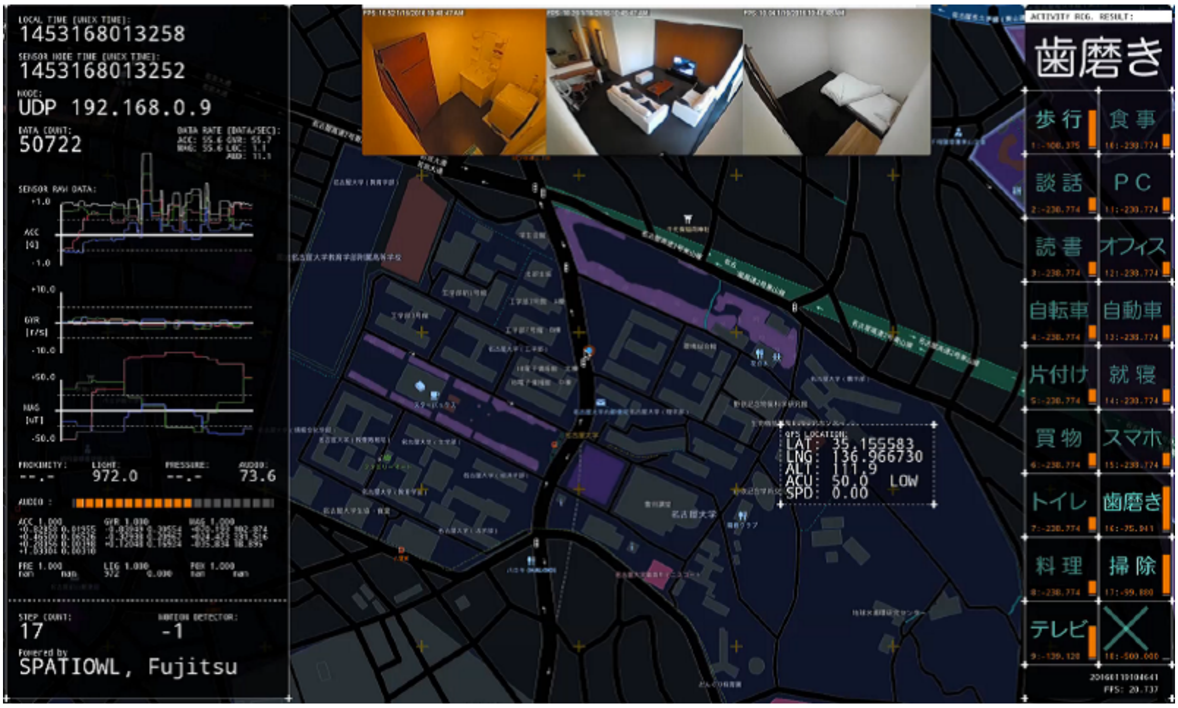

- Discovery of new knowledge
- Sense of accomplishment
Experience
Activity
Ability
- Intellectual activity : study, communication, etc.
- Physical activity : hobby, sports, etc.
- Intelligence
- Physical fitness

Figure 6.2: *Targeted cycle including human activity recognition. Activities yield experiences, such as the discovery of new knowledge or a sense of accomplishment. These experiences enhance people's abilities, expanding the range of activities that can be attempted.*

cooking. There are two important points to be considered when developing such applications; the first is the usability of the system, and the second is the range of recognizable activities under realistic conditions. To address the first point, a smartphone-based recording system was used to collect data, and this system does not require a large number of sensors and is easy to use. To address the second point, a large database of human activity consisting of multi-modal signals recorded under realistic conditions, called the Nagoya-COI daily activity database, was created, and two deep neural network (DNN)-based fusion methods that use multi-modal signals to recognize complex human activities were proposed. Furthermore, speaker adaptation techniques were introduced to address the problem of model individuality, which results in degradation in performance when a model constructed for a particular subject is used to classify the activities of another individual. Experimental results using the constructed database demonstrated that the use of multi-modal signals is effective, and that speaker adaptation techniques can improve activity recognition performance, especially when using only a limited amount of training data. Moreover, a human activity visualization tool was developed, and is shown in Fig. 6.3. In Fig. 6.3, the right side represents the results of activity recognition, and the left side displays the recorded signals, while the center shows the monitored video and the geographic location of the smartphone user. This visualization tool can be used to facilitate RWDC as it occurs in actual human lives.

## 6.3 Polyphonic SED based on Duration Modeling

In order to realize practical applications, characteristics of acquired data need to be analyzed in detail and a method based on these characteristics will need to be developed. The work for SED is a good illustration of the analysis phase of RWDC. In this work, modeling the duration of sound events was the main focus. To do this, a new hybrid approach using duration-controlled long short-term memory (LSTM) was proposed, which builds upon a



LOCAL TIME (UNIX TIME):
1453168013258

SENSOR NODE TIME (UNIX TIME):
1453168013252

NODE:
UDP 192.168.0.9

DATA COUNT:
50722

DATA RATE (DATA/SEC):
ACC: 55.6 GYR: 55.7
MAG: 55.6 LOC: 1.1
AUD: 11.1

PROXIMITY: --.-
LIGHT: 972.0
PRESSURE: --.-
AUDIO: 73.6

STEP COUNT:
17

MOTION DETECTOR:
-1

Powered by
SPATIOWL, Fujitsu

GPS LOCATION:
LAT: 35.155583
LNG: 136.966730
ALT: 111.9
ACU: 50.0 LOW
SPD: 0.00

ACTIVITY REG. RESULT:
歯磨き

20160119104641
FPS: 20.737

Figure 6.3: *Human activity visualization tool. On the right, the results of activity recognition are shown. On the left, the recorded signals are displayed. A monitored video is shown at the top center, and the location of the subject with the smartphone is indicated on the map in the center.*

state-of-the-art SED method that performs frame-by-frame detection using a bidirectional LSTM recurrent neural network (BLSTM) by incorporating a duration-controlled modeling technique based on a hidden Markov model (HMM) or a hidden semi-Markov model (HSMM). The proposed approach made it possible to model the duration of each sound event precisely and to perform sequence-by-sequence detection without any need for thresholding. Furthermore, to effectively reduce insertion errors which often occur under noisy conditions, a post-processing step based binary masks on was also introduced, which relies on a sound activity detection (SAD) network to identify segments with an arbitrary sound event activity. Experimental evaluation with the DCASE2016 task2 dataset [4] demonstrated that the proposed method outperformed conventional polyphonic SED methods, proving that modeling of sound event duration is effective for polyphonic SED. This work therefore resulted in the successful development of a new method of polyphonic SED through the analysis of data acquired from the real world.

## 6.4 Anomalous SED based on Waveform Modeling

To realize practical applications using machine learning techniques, it is necessary to consider how to collect and annotate date. In the case of SED, these annotations include the start and the end timestamps of each sound event and the label for them. However, since sound events include a wide range of sounds that vary greatly in their acoustic characteristics, duration, and volume, the data is difficult to annotate accurately, even for humans. To address this issue, a new SED method that operates in an unsupervised manner was proposed, making it possible to detect novel sound events that do not appear in training data. The normal acoustic patterns occurring in public spaces in the time domain were modeled using WaveNet, which is a generative model based on an autoregressive convolutional neural network (CNN). The proposed method uses WaveNet as a predictor rather than as a generator to detect the waveform segments responsible for the large prediction errors as novel acoustic patterns. Since the novel detected patterns were not included in the training data, they could be used to train standard SED systems to improve their performance. Therefore, a data circulation process involving data acquisition, analysis, and theoretical implementation was realized by combining the proposed method with other SED systems.

## 6.5 Summary

This chapter discussed the relation of each work in this thesis research to RWDC. Through the work on human activity recognition based on DNN using multi-modal signals, data circulation intended to enhance the quality of life was described. Then, in the work on polyphonic SED based on duration modeling, a new method of SED was successfully developed by analyzing the characteristics of the acquired data. Finally, in the work on anomalous SED based on waveform modeling, the cycle of data acquisition, analysis and implementation was demonstrated by applying the output of the proposed anomalous SED method in order to improve performance.

# 7 Conclusions

## 7.1 Summary of the Thesis

Sound event detection (SED), which is the task to detect the beginning and the end times of sound events and to label them, has great potential for various applications. SED is a challenging task since sound events include a wide range of phenomena, which vary widely in their acoustic characteristics, duration, and volume. Though recent advances in machine learning techniques have improved the performance of SED systems, various problems remain to be solved.

This thesis addressed the following three problems. The first is how to combine different types of signals to extend the range of detectable sound events. The second is how to model the duration of sound events to improve polyphonic SED performance. And the third is how to model normal environments in order to improve anomalous SED performance.

Chapter 2 provided an overview of the field of environmental sound understanding and its applications. Four major tasks that researchers have focused on in the field of environmental sound understanding were introduced, and various techniques that have been used to perform each task were also reviewed.

In Chapter 3, toward the development of smartphone-based monitoring system for life logging, two neural network-based fusion methods with multi-modal signals were proposed and then a large-scale human activity database was created for testing. This database includes over 1,400 hours of data including both outdoor and indoor daily activities recorded by 19 subjects in practical situations. Furthermore, speaker adaptation techniques in the field of automatic speech recognition (ASR) were introduced to address the problem of model individuality. Experimental results using the constructed database showed that the use of multimodal data including environmental sound and acceleration signals with neural networks is effective, and that adaptation methods improved performance, especially when using only a limited amount of subject-specific training data.

In Chapter 4, a new hybrid approach for polyphonic SED called duration-controlled long short-term memory (LSTM) was proposed, which incorporates a duration-controlled modeling technique based on a hidden semi-Markov model (HSMM) and a frame-by-frame detection method based on a bidirectional LSTM (BLSTM). The proposed duration-controlled LSTM made it possible to model the duration of each sound event explicitly and to perform sequence-by-sequence detection without any need for thresholding. Furthermore, to reduce insertion errors that tend to often occur under noisy conditions, post-processing step using a sound activity detection (SAD) network was also used to identify segments with any kind of sound event activity. The proposed method outperformed the best results submitted for the DCASE2016 task2 challenge, proving that modeling sound event duration is effective for polyphonic SED. Furthermore, combining the three BLSTM-based methods allowed further improvements.

In Chapter 5, a new anomalous SED method that uses WaveNet was proposed to directly model various acoustic patterns in the time domain. The proposed method used WaveNet as a predictor rather than as a generator to detect waveform segments that cause large prediction errors as anomalous acoustic patterns. Because WaveNet is capable of modeling detailed temporal structures, such as the phase information of the waveform signals, the proposed method can detect anomalous sound events more accurately than conventional methods based on reconstruction errors of acoustic features. Furthermore, to consider differences in environmental situations, i-vector was used as an additional auxiliary feature of WaveNet. The i-vector extractor allowed to discriminate sound patterns, depending on the time, location, and surrounding environment. Experimental evaluation using a database recorded in public spaces showed that the proposed method outperformed conventional feature-based approaches, and that modeling in the time domain in conjunction with the i-vectors extractor is effective for anomalous SED.

Chapter 6 introduced the concept and importance of real world data circulation (RWDC), which is a key for the success of a commercial application. Furthermore, each work in this thesis was discussed in terms of RWDC. This chapter also described that how the developed tool for human activity recognition could be used to foster RWDC, how the analysis of data during research on polyphonic SED inspired a new method based on duration modeling, and how outputs from the research on anomalous SED could be used to improve the performance of polyphonic SED systems.

## 7.2 Future Work

Although the proposed methods have improved the performance of SED systems, a number of challenges still remain.

**Investigation of the effect of recording conditions:** In the field of ASR, performance degrades significantly in recording environments that include background noise and reverberation. However, in the research described above, the performance of the proposed methods under different recording conditions was not investigated. To apply the proposed techniques in practical applications, the proposed methods will be needed to be tested with data collected under a range for recording conditions.

**Further extension of the range of detectable events:** Chapter 3 described how the system extends the range of detectable events with multi-modal signals. However, all of the detectable events were indoor activities. To make life-logging systems more useful, the proposed method will need to detect outdoor activities as well. To address this issue, the use of additional types of signals should be considered.

**Utilization of multi-channel information:** Recent improvements in recording equipment have made it easy to record multi-channel signals that contain a variety of useful information like the direction of arrival. Various front-end processing techniques using multi-channel information have been proposed which enhance or separate target signals from a mixture of signals [152–154]. These techniques should be considered to improve the performance of the proposed methods under noisy conditions.

**End-to-end processing:** The effectiveness of the proposed SED systems was demonstrated, however, these approaches consist of several modules and steps, and each module needs to be optimized to achieve good performance. In the fields of ASR and text-to-speech (TTS), end-to-end models that can complete all processing within a single neural network allow the jointly optimization of the entire system, achieving superior performance [138, 155–158]. Therefore, the performance of the SED systems can be further improved by replacing each module with a neural network that can then be integrated into a single neural network.

**Utilization of ontology information:** One of the problems faced by current SED systems is the varying granularity of sound-event labels. In the case of ASR or TTS, for example, the minimum unit of target labeling is predefined (e.g., a phoneme or a character), and all the target labels can be decomposed into subsets of that unit. SED target labels have no minimum unit, however, so labels with different granularities must serve as the targets. Ontology information, i.e., information about the relationships among the labels, may be useful for addressing this issue. In the field of object detection, hierarchical classification using ontology information has been proposed [159], which allows classifiers to deal with labels with varied granularities. More research is needed about utilizing ontology information to improve the performance of SED systems.

# Acknowledgments

I would like to express my deepest appreciation to my thesis advisor, Professor Kazuya Takeda of Nagoya University, for his constant guidance and encouragement throughout my bachelor's, master's and doctorate courses.

I would also like to express my gratitude to Professor Tomoki Toda of Nagoya University for his insightful suggestions and continuous support throughout my master's and doctorate courses. He has taught me many important things, from how to write a paper to supporting my motivation for research. This work could not have been accomplished without his direction.

I would like to express my sincere and deep appreciation to Associate Research Professor Shinji Watanabe of Johns Hopkins University for giving me the opportunity to work as an Intern Researcher at Mitsubishi Electric Research Laboratories, and for his continued discussion about various research topics. Thanks to him, I have been able to expand the scope of my research and have enjoyed many opportunities to improve my skills. I could not have achieved my research goals without his support and encouragement.

I would like to express my sincere gratitude to Associate Professor Masafumi Nishida of Shizuoka University for his helpful advice and suggestions throughout my master's course. One of the core sections of this thesis could not have been accomplished without his direction and support.

I would especially like to note my tremendous appreciation to Professor Norihide Kitaoka

of Tokushima University for his continuous support throughout my bachelor's and master's courses. Thanks to his help and encouragement, I have mastered the fundamentals of scientific research.

I would like to express my extreme gratitude to Senior Researcher Tatsuya Komatsu of NEC Corporation for giving me the opportunity to work as an Intern Researcher at NEC's Data Science Laboratories. His useful advices about research and other subjects, encouraged me to persevere with my studies.

I would also like to express my gratitude to Professor Takanori Nishino of Meijo University, Professor Hiromi Nakaiwa of Nagoya University, Associate Professor Yoshio Ishiguro of Nagoya University, Associate Professor Chiyomi Miyajima of Daido University, Associate Professor Eijiro Takeuchi of Nagoya University, Associate Professor Takatoshi Jitsuhiro of Aichi University of Technology, Assistant Professor Alexander Carballo Segura of Nagoya University, Assistant Professor Reo Iijima of Nagoya University, and Dr. Akira Tamamori of the Aichi Institute of Technology, for their valuable comments and suggestions.

I would like to acknowledge and thank the staffs of Takeda Laboratory, Toda Laboratory, and RWDC Leading Programs for their kind support. I would also like to thank my laboratory colleagues for their support, especially Dr. Kento Ohtani of Human Dataware Lab. Co., Ltd., for his generous advice and support in the lab. I could not have completed my studies without his support.

Finally, I would like to express my indebtedness and my thanks to my family for their understanding, support and encouragement throughout my university studies.

# References

[1] R. Stiefelhagen, K. Bernardin, R. Bowers, R. T. Rose, M. Michel, and J. Garofolo, "The CLEAR 2007 evaluation," in *Multimodal Technologies for Perception of Humans*. Springer, 2008, pp. 3–34.

[2] TREC Video Retrieval Evaluation: TRECVID. http://www-nlpir.nist.gov/projects/trecvid/.

[3] Detection and Classification of Acoustic Scenes and Events 2013 - DCASE 2013. http://c4dm.eecs.qmul.ac.uk/sceneseventschallenge/.

[4] Detection and Classification of Acoustic Scenes and Events 2016 - DCASE 2016. http://www.cs.tut.fi/sgn/arg/dcase2016/.

[5] Detection and Classification of Acoustic Scenes and Events 2017 - DCASE 2017. http://www.cs.tut.fi/sgn/arg/dcase2017/.

[6] DCASE2018 Challenge. http://dcase.community/challenge2018/.

[7] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proc. ACM International Conference on Multimedia*, Orlando, FL, USA, Nov. 2014, pp. 1041–1044.

[8] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio set: An ontology and human-labeled dataset for audio events,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2017, pp. 776–780.

[9] M. Xu, C. Xu, L. Duan, J. S. Jin, and S. Luo, “Audio keywords generation for sports video analysis,” *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 4, no. 2, p. 11, 2008.

[10] Y.-T. Peng, C.-Y. Lin, M.-T. Sun, and K.-C. Tsai, “Healthcare audio event classification using hidden Markov models and hierarchical hidden Markov models,” in *Proc. IEEE International Conference on Multimedia and Expo*. IEEE, 2009, pp. 1218–1221.

[11] J. A. Stork, L. Spinello, J. Silva, and K. O. Arras, “Audio-based human activity recognition using non-Markovian ensemble voting,” in *Proc. IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 2012, pp. 509–514.

[12] D. Valtchev and I. Frankov, “Service gateway architecture for a smart home,” *IEEE Communications Magazine*, vol. 40, no. 4, pp. 126–132, 2002.

[13] C. Clavel, T. Ehrette, and G. Richard, “Events detection for an audio-based surveillance system,” in *Proc. IEEE International conference on Multimedia and Expo*. IEEE, 2005, pp. 1306–1309.

[14] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, “Scream and gunshot detection and localization for audio-surveillance systems,” in *Proc. IEEE Conference on Advanced Video and Signal Based Surveillance*. IEEE, 2007, pp. 21–26.

[15] Y. Chung, S. Oh, J. Lee, D. Park, H.-H. Chang, and S. Kim, “Automatic detection and recognition of pig wasting diseases using sound data in audio surveillance systems,” *Sensors*, vol. 13, no. 10, pp. 12 929–12 942, 2013.

[16] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “WaveNet: A generative model for raw audio,” *CoRR abs/1609.03499*, 2016.

[17] Graduate Program for Real-World Data Circulation Leaders. http://www.rwdc.is.nagoya-u.ac.jp/.

[18] E. C. Cherry, “Some experiments on the recognition of speech, with one and with two ears,” *The Journal of the acoustical society of America*, vol. 25, no. 5, pp. 975–979, 1953.

[19] B. Arons, “A review of the cocktail party effect,” *Journal of the American Voice I/O Society*, vol. 12, no. 7, pp. 35–50, 1992.

[20] A. R. Conway, N. Cowan, and M. F. Bunting, “The cocktail party phenomenon revisited: The importance of working memory capacity,” *Psychonomic bulletin & review*, vol. 8, no. 2, pp. 331–335, 2001.

[21] G. J. Brown and M. Cooke, “Computational auditory scene analysis,” *Computer Speech & Language*, vol. 8, no. 4, pp. 297–336, 1994.

[22] A. S. Bregman, *Auditory scene analysis: The perceptual organization of sound.* MIT press, 1994.

[23] D. P. W. Ellis, “Prediction-driven computational auditory scene analysis,” Ph.D. dissertation, Massachusetts Institute of Technology, 1996.

[24] D. F. Rosenthal and H. G. Okuno, *Computational auditory scene analysis.* Lawrence Erlbaum Associates Publishers, 1998.

[25] D. Wang and G. J. Brown, *Computational auditory scene analysis: Principles, algorithms, and applications*. Wiley-IEEE press, 2006.

[26] Y. Oishi, "Toward detection and discrimination of all sounds: present and future of audio event detection," *Spring Meeting of Acoustical Scoriety of Japan*, pp. 1521–1524, 2014.

[27] J. Ramirez, J. M. Górriz, and J. C. Segura, "Voice activity detection. fundamentals and speech recognition system robustness," in *Robust speech recognition and understanding*. InTech, 2007.

[28] D. A. Reynolds, "An overview of automatic speaker recognition technology," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2002, pp. 4072–4075.

[29] A. J. Eronen, V. T. Peltonen, J. T. Tuomi, A. P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, "Audio-based context recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 321–329, 2006.

[30] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *Proc. IEEE Workshop on Mobile Computing Systems and Applications*. IEEE, 1994, pp. 85–90.

[31] M. Haggblade, Y. Hong, and K. Kao, "Music genre classification," *Department of Computer Science, Stanford University*, 2011.

[32] Y. Panagakis, C. L. Kotropoulos, and G. R. Arce, “Music genre classification via joint sparse low-rank representation of audio features,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1905–1917, 2014.

[33] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[34] S. Pancoast and M. Akbacak, “N-gram extension for bag-of-audio-words,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 778–782.

[35] E. Amid, A. Mesaros, K. J. Palomaki, J. Laaksonen, and M. Kurimo, “Unsupervised feature extraction for multimedia event detection and ranking using audio content,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2014, pp. 5939–5943.

[36] J. T. Geiger, B. Schuller, and G. Rigoll, “Large-scale audio feature extraction and SVM for acoustic scene classification,” in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2013, pp. 1–4.

[37] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer, “CP-JKU submissions for DCASE-2016: a hybrid approach using binaural i-vectors and deep convolutional neural networks,” DCASE2016 Challenge, Tech. Rep., September 2016.

[38] B. Lehner, H. Eghbal-Zadeh, M. Dorfer, F. Korzeniowski, K. Koutini, and G. Widmer, “Classifying short acoustic scenes with I-vectors and CNNs: Challenges and optimisations for the 2017 DCASE ASC task,” DCASE2017 Challenge, Tech. Rep., September 2017.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. Advances in neural information processing systems*, 2012, pp. 1097–1105.

[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[41] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[42] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” *CoRR*, vol. abs/1610.02357, 2016. [Online]. Available: http://arxiv.org/abs/1610.02357

[43] Y. Han and J. Park, “Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification,” DCASE2017 Challenge, Tech. Rep., September 2017.

[44] Y. Sakashita and M. Aono, “Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions,” DCASE2018 Challenge, Tech. Rep., September 2018.

[45] M. Dorfer, B. Lehner, H. Eghbal-zadeh, H. Christop, P. Fabian, and W. Gerhard, “Acoustic scene classification with fully convolutional neural networks and I-vectors,” DCASE2018 Challenge, Tech. Rep., September 2018.

[46] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, “Very deep convolutional neural networks for raw waveforms,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2017, pp. 421–425.

[47] J. Lee, T. Kim, J. Park, and J. Nam, “Raw waveform-based audio classification using sample-level CNN architectures,” *CoRR*, vol. abs/1712.00866, 2017. [Online]. Available: http://arxiv.org/abs/1712.00866

[48] J. Salamon and J. P. Bello, “Deep convolutional neural networks and data augmentation for environmental sound classification,” *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[49] Y. Han and K. Lee, “Acoustic scene classification using convolutional neural network and multiple-width frequency-delta data augmentation,” *arXiv preprint arXiv:1607.02383*, 2016.

[50] J. Schröder, B. Cauchi, M. R. Schädler, N. Moritz, K. Adiloglu, J. Anemüller, S. Doclo, B. Kollmeier, and S. Goetze, “Acoustic event detection using signal enhancement and spectro-temporal feature extraction,” in *Proc. IEEE Workshop Applications of Signal Processing to Audio and Acoustics*, 2013.

[51] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, “Context-dependent sound event detection,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, no. 1, pp. 1–13, 2013.

[52] T. Heittola, A. Mesaros, T. Virtanen, and A. Eronen, “Sound event detection in multisource environments using source separation,” in *Proc. Workshop on Machine Listening in Multisource Environments*, 2011, pp. 36–40.

[53] S. Innami and H. Kasai, “NMF-based environmental sound source separation using time-variant gain features,” *Computers & Mathematics with Applications*, vol. 64, no. 5, pp. 1333–1342, 2012.

[54] A. Dessein, A. Cont, and G. Lemaitre, “Real-time detection of overlapping sound events with non-negative matrix factorization,” in *Matrix Information Geometry*. Springer, 2013, pp. 341–371.

[55] T. Komatsu, T. Toizumi, R. Kondo, and Y. Senda, “Acoustic event detection method using semi-supervised non-negative matrix factorization with mixtures of local dictionaries,” in *Proc. the Detection and Classification of Acoustic Scenes and Events 2016 Workshop*, September 2016, pp. 45–49.

[56] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, “Polyphonic sound event detection using multi label deep neural networks,” in *Proc. International Joint Conference on Neural Networks*. IEEE, 2015, pp. 1–7.

[57] G. Parascandolo, H. Huttunen, and T. Virtanen, “Recurrent neural networks for polyphonic sound event detection in real life recordings,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2016, pp. 6440–6444.

[58] M. Espi, M. Fujimoto, K. Kinoshita, and T. Nakatani, “Exploiting spectro-temporal locality in deep learning based acoustic event detection,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, no. 1, p. 1, 2015.

[59] Y. Wang, L. Neves, and F. Metze, “Audio-based multimedia event detection using deep recurrent neural networks,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2016, pp. 2742–2746.

[60] F. Eyben, S. Böck, B. Schuller, A. Graves *et al.*, “Universal onset detection with bidirectional long short-term memory neural networks.” in *Proc. International Society for Music Information Retrieval Conference*, 2010, pp. 589–594.

[61] I. Choi, K. Kwon, S. H. Bae, and N. S. Kim, "DNN-based sound event detection with exemplar-based approach for noise reduction," in *Proc. the Detection and Classification of Acoustic Scenes and Events 2016 Workshop*, September 2016, pp. 16–19.

[62] S. Adavanne, G. Parascandolo, P. Pertila, T. Heittola, and T. Virtanen, "Sound event detection in multichannel audio using spatial and harmonic features," in *Proc. the Detection and Classification of Acoustic Scenes and Events 2016 Workshop*, September 2016, pp. 6–10.

[63] L. JiaKai, "Mean teacher convolution system for DCASE 2018 task 4," DCASE2018 Challenge, Tech. Rep., September 2018.

[64] Y. L. Liu, J. Yan, Y. Song, and J. Du, "USTC-NELSLIP system for DCASE 2018 challenge task 4," DCASE2018 Challenge, Tech. Rep., September 2018.

[65] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[66] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Proc. International Conference on Pervasive Computing*. Springer, 2004, pp. 1–17.

[67] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford, "A hybrid discriminative/generative approach for modeling human activities," 2005.

[68] T. Huynh and B. Schiele, "Towards less supervision in activity recognition from wearable sensors," in *Proc. IEEE International Symposium on Wearable Computers*. IEEE, 2006, pp. 3–10.

[69] K. Ouchi and M. Doi, “Smartphone-based monitoring system for activities of daily living for elderly people and their relatives etc.” in *Proc. ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*. ACM, 2013, pp. 103–106.

[70] T. Maekawa, Y. Yanagisawa, Y. Kishino, K. Ishiguro, K. Kamei, Y. Sakurai, and T. Okadome, “Object-based activity recognition with heterogeneous sensors on wrist,” in *Proc. International Conference on Pervasive Computing*. Springer, 2010, pp. 246–264.

[71] P. Lukowicz, J. A. Ward, H. Junker, M. Stäger, G. Tröster, A. Atrash, and T. Starner, “Recognizing workshop activity using body worn microphones and accelerometers,” in *Proc. International Conference on Pervasive Computing*. Springer, 2004, pp. 18–32.

[72] T. Bertin-Mahieux, D. Eck, and M. Mandel, “Automatic tagging of audio: The state-of-the-art,” in *Machine audition: Principles, algorithms and systems*. IGI Global, 2011, pp. 334–352.

[73] P. Foster, S. Sigtia, S. Krstulovic, J. Barker, and M. D. Plumbley, “Chime-home: A dataset for sound source recognition in a domestic environment.” in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2015, pp. 1–5.

[74] T. Lidy and A. Schindler, “CQT-based convolutional neural networks for audio scene classification and domestic audio tagging,” DCASE2016 Challenge, Tech. Rep., September 2016.

[75] Y. Xu, Q. Huang, W. Wang, P. Foster, S. Sigtia, P. J. Jackson, and M. D. Plumbley, “Unsupervised feature learning based on deep models for environmental audio tag-

ging," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1230–1241, 2017.

[76] M. Basseville, I. V. Nikiforov *et al.*, *Detection of abrupt changes: theory and application*. Prentice Hall Englewood Cliffs, 1993, vol. 104.

[77] S. Liu, M. Yamada, N. Collier, and M. Sugiyama, "Change-point detection in time-series data by relative density-ratio estimation," *Neural Networks*, vol. 43, pp. 72–83, 2013.

[78] Y. Ono, Y. Onishi, T. Koshinaka, S. Takata, and O. Hoshuyama, "Anomaly detection of motors with feature emphasis using only normal sounds," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 2800–2804.

[79] L. Tarassenko, P. Hayton, N. Cerneaz, and M. Brady, "Novelty detection for the identification of masses in mammograms," 1995.

[80] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proc. IEEE International Conference on Multimedia and Expo*, vol. 1. IEEE, 2000, pp. 452–455.

[81] M. Markou and S. Singh, "Novelty detection: a review part 1: statistical approaches," *Signal processing*, vol. 83, no. 12, pp. 2481–2497, 2003.

[82] D. W. Scott, "Outlier detection and clustering by partial mixture modeling," in *Proc. Computational Statistics*. Springer, 2004, pp. 453–464.

[83] J. Ma and S. Perkins, "Time-series novelty detection using one-class support vector machines," in *Proc. IEEE International Joint Conference on Neural Networks*, vol. 3. IEEE, 2003, pp. 1741–1745.

[84] M. Sakurada and T. Yairi, “Anomaly detection using autoencoders with nonlinear dimensionality reduction,” in *Proc. ACM Workshop on Machine Learning for Sensory Data Analysis*. ACM, 2014, p. 4.

[85] E. Marchi, F. Vesperini, F. Eyben, S. Squartini, and B. Schuller, “A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2015, pp. 1996–2000.

[86] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, “Long short term memory networks for anomaly detection in time series,” in *Proceedings*. Presses universitaires de Louvain, 2015, p. 89.

[87] C. Gurrin, A. F. Smeaton, A. R. Doherty *et al.*, “Lifelogging: Personal big data,” *Foundations and Trends® in Information Retrieval*, vol. 8, no. 1, pp. 1–125, 2014.

[88] M. P. Rajasekaran, S. Radhakrishnan, and P. Subbaraj, “Elderly patient monitoring system using a wireless sensor network,” *Telemedicine and e-Health*, vol. 15, no. 1, pp. 73–79, 2009.

[89] Q. Lin, D. Zhang, X. Huang, H. Ni, and X. Zhou, “Detecting wandering behavior based on GPS traces for elders with dementia,” in *Proc. IEEE Conference on Control Automation Robotics & Vision*. IEEE, 2012, pp. 672–677.

[90] Y. Liang, X. Zhou, Z. Yu, and B. Guo, “Energy-efficient motion related activity recognition on mobile devices for pervasive healthcare,” *Mobile Networks and Applications*, vol. 19, no. 3, pp. 303–317, 2014.

[91] P. Rashidi and D. J. Cook, "Keeping the resident in the loop: Adapting the smart home to the user," *IEEE Transactions on Systems, Man, and Cybernetics-part A: Systems and Humans*, vol. 39, no. 5, pp. 949–959, 2009.

[92] C. Zhang and Y. Tian, "RGB-D camera-based daily living activity recognition," *Journal of Computer Vision and Image Processing*, vol. 2, no. 4, p. 12, 2012.

[93] A. Iosifidis, A. Tefas, and I. Pitas, "View-invariant action recognition based on artificial neural networks," *Proc. IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 3, pp. 412–424, 2012.

[94] Z. Deng, A. Vahdat, H. Hu, and G. Mori, "Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2016, pp. 4772–4781.

[95] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori, "A hierarchical deep temporal model for group activity recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2016, pp. 1971–1980.

[96] D. J. Patterson, D. Fox, H. Kautz, and M. Philipose, "Fine-grained activity recognition by aggregating abstract object usage," in *Proc. IEEE International Symposium on Wearable Computers*. IEEE, 2005, pp. 44–51.

[97] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg, "A scalable approach to activity recognition based on object use," in *Proc. IEEE International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[98] M. Philipose, K. P. Fishkin, M. Perkowitz, D. J. Patterson, D. Fox, H. Kautz, and D. Hahnel, "Inferring activities from interactions with objects," *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 50–57, 2004.

[99] A. Fleury, M. Vacher, and N. Noury, "SVM-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 2, pp. 274–283, 2010.

[100] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Proc. International Symposium on Wearable Computers*. IEEE, 2012, pp. 108–109.

[101] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. d. R. Millán, and D. Roggen, "The opportunity challenge: A benchmark database for on-body sensor-based activity recognition," *Pattern Recognition Letters*, vol. 34, no. 15, pp. 2033–2042, 2013.

[102] H. Guo, L. Chen, L. Peng, and G. Chen, "Wearable sensor based multimodal human activity recognition exploiting the diversity of classifier ensemble," in *Proc. ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 1112–1123.

[103] F. J. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[104] N. Y. Hammerla, S. Halloran, and T. Ploetz, “Deep, convolutional, and recurrent models for human activity recognition using wearables,” *CoRR*, vol. abs/1604.08880, 2016. [Online]. Available: http://arxiv.org/abs/1604.08880

[105] ELAN-Linguistic Annotator. http://www.mpi.nl/corpus/html/elan.

[106] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[107] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *CoRR*, vol. abs/1207.0580, 2012. [Online]. Available: http://arxiv.org/abs/1207.0580

[108] T. Ochiai, S. Matsuda, X. Lu, C. Hori, and S. Katagiri, “Speaker adaptive training using deep neural networks,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2014, pp. 6349–6353.

[109] T. Ochiai, S. Matsuda, H. Watanabe, X. Lu, C. Hori, and S. Katagiri, “Speaker adaptive training for deep neural networks embedding linear transformation networks,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2015, pp. 4605–4609.

[110] Torch 7 — A Scientific Computing Framework for Luajit. http://torch.ch/.

[111] C.-C. Chang and C.-J. Lin, “LIBSVM: a library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, p. 27, 2011.

[112] L. v. d. Maaten and G. Hinton, “Visualizing data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[113] A. Harma, M. F. McKinney, and J. Skowronek, "Automatic surveillance of the acoustic activity in our living environment," in *Proc. IEEE International Conference on Multimedia and Expo*. IEEE, 2005, pp. 4–8.

[114] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Audio context recognition using audio event histograms," in *Proc. European Signal Processing Conference*. IEEE, 2010, pp. 1272–1276.

[115] L. M. Aiello, R. Schifanella, D. Quercia, and F. Aletta, "Chatty maps: constructing sound maps of urban areas from social media data," *Open Science*, vol. 3, no. 3, p. 150690, 2016.

[116] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling." in *Proc. Interspeech*, 2014, pp. 338–342.

[117] Z. Chen, S. Watanabe, H. Erdogan, and J. Hershey, "Integration of speech enhancement and recognition using long-short term memory recurrent neural network," in *Proc. Interspeech*, 2015.

[118] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 6645–6649.

[119] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters*, vol. 6, no. 1, pp. 1–3, 1999.

[120] J. Ramırez, J. C. Segura, C. Benıtez, A. De La Torre, and A. Rubio, "Efficient voice activity detection algorithms using long-term speech information," *Speech Communication*, vol. 42, no. 3, pp. 271–287, 2004.

[121] M. Fujimoto, K. Ishizuka, and T. Nakatani, "A voice activity detection based on the adaptive integration of multiple speech features and a signal decision scheme," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 4441–4444.

[122] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[123] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[124] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[125] S. Z. Yu, "Hidden semi-Markov models," *Artificial Intelligence*, vol. 174, no. 2, pp. 215–243, 2010.

[126] Z. Heiga, K. Tokuda, T. Masuko, T. Kobayasih, and T. Kitamura, "A hidden semi-Markov model-based speech synthesis system," *IEICE Transactions on Information and Systems*, vol. 90, no. 5, pp. 825–834, 2007.

[127] HTK Speech Recognition Toolkit. http://htk.eng.cam.ac.uk.

[128] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[129] R. Caruana, "Multitask learning," in *Learning to learn*. Springer, 1998, pp. 95–133.

[130] M. L. Seltzer and J. Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 6965–6969.

[131] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. ACM International Conference on Machine Learning*. ACM, 2008, pp. 160–167.

[132] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

[133] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "TensorFlow: A system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[134] S. C. Lee and R. Nevatia, "Hierarchical abnormal event detection by real time and semi-real time multi-tasking video surveillance system," *Machine vision and applications*, vol. 25, no. 1, pp. 133–143, 2014.

[135] R. Arroyo, J. J. Yebes, L. M. Bergasa, I. G. Daza, and J. Almazán, "Expert video-surveillance system for real-time detection of suspicious behaviors in shopping malls," *Expert systems with Applications*, vol. 42, no. 21, pp. 7991–8005, 2015.

[136] S. Adavanne, A. Politis, and T. Virtanen, "Multichannel sound event detection using 3d convolutional neural networks for learning inter-channel features," *CoRR*, vol. abs/1801.09522, 2018. [Online]. Available: http://arxiv.org/abs/1801.09522

[137] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-dependent WaveNet vocoder," in *Proc. Interspeech*, vol. 2017, 2017, pp. 1118–1122.

[138] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions," *CoRR*, vol. abs/1712.05884, 2017. [Online]. Available: http://arxiv.org/abs/1712.05884

[139] G. Recommendation, "Pulse code modulation (PCM) of voice frequencies," *ITU*, 1988.

[140] T. L. Paine, P. Khorrami, S. Chang, Y. Zhang, P. Ramachandran, M. A. Hasegawa-Johnson, and T. S. Huang, "Fast WaveNet generation algorithm," *CoRR*, vol. abs/1611.09482, 2016. [Online]. Available: http://arxiv.org/abs/1611.09482

[141] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis, "Parallel WaveNet: Fast high-fidelity speech synthesis," *CoRR*, vol. abs/1711.10433, 2017. [Online]. Available: http://arxiv.org/abs/1711.10433

[142] T. Hayashi, A. Tamamori, K. Kobayashi, K. Takeda, and T. Toda, "An investigation of multi-speaker training for WaveNet vocoder," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2017, pp. 712–718.

[143] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[144] C. J. Leggetter and P. C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models,” *Computer speech & language*, vol. 9, no. 2, pp. 171–185, 1995.

[145] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, “SVM based speaker verification using a GMM supervector kernel and NAP variability compensation,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1. IEEE, 2006, pp. I–I.

[146] A. O. Hatch, S. Kajarekar, and A. Stolcke, “Within-class covariance normalization for svm-based speaker recognition,” in *Proc. Ninth international conference on spoken language processing*, 2006.

[147] D. Garcia-Romero and C. Y. Espy-Wilson, “Analysis of i-vector length normalization in speaker recognition systems,” in *Proc. Annual Conference of the International Speech Communication Association*, 2011.

[148] A. Kanagasundaram, D. B. Dean, R. Vogt, M. Mclaren, S. Sridharan, and M. W. Mason, “Weighted LDA techniques for i-vector based speaker verification,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2012, pp. 4781–4784.

[149] Series 6000 general sound effects library. http://www.sound-ideas.com/sound-effects/series-6000-sound-effects-library.html.

[150] F. Chollet *et al.* (2015) Keras. https://github.com/fchollet/keras.

[151] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, “The kaldi speech recognition toolkit,”

in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[152] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 550–563, 2010.

[153] D. Kitamura, N. Ono, H. Sawada, H. Kameoka, and H. Saruwatari, "Determined blind source separation unifying independent vector analysis and nonnegative matrix factorization," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 9, pp. 1622–1637, 2016.

[154] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2016, pp. 196–200.

[155] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. International Conference on Machine Learning*, 2014, pp. 1764–1772.

[156] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. Advances in neural information processing systems*, 2015, pp. 577–585.

[157] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[158] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. V. Le, Y. Agiomyrgiannakis, R. Clark, and R. A.

Saurous, “Tacotron: A fully end-to-end text-to-speech synthesis model,” *CoRR*, vol. abs/1703.10135, 2017. [Online]. Available: http://arxiv.org/abs/1703.10135

[159] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” *CoRR*, vol. abs/1612.08242, 2016. [Online]. Available: http://arxiv.org/abs/1612.08242

# List of Publications

## Journal Papers

1. T. Hayashi, S. Watanabe, T. Toda, T. Hori, J. Le Roux, K. Takeda, “Duration-controlled LSTM for polyphonic sound event detection,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 25, No. 11, pp. 2059–2070, Nov. 2017.

2. T. Hayashi, M. Nishida, N. Kitaoka, T. Toda, K. Takeda, “Daily activity recognition with large-scaled real-life recording datasets based on deep neural network using multi-modal signals,” IEICE Transactions on Fundamentals, Vol. E101-A, No. 1, pp. 199–210, Jan. 2018.

3. Watanabe, T. Hori, S. Kim, J. R. Hershey, T. Hayashi, “Hybrid CTC/Attention architecture for end-to-end speech recognition,” in IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 8, pp. 1240–1253, Dec. 2017.

4. A. Tamamori, T. Hayashi, T. Toda, K. Takeda, “Daily activity recognition based on recurrent neural network using multi-modal signals,” APSIPA Transactions on Signal and Information Processing (accepted).

# International Conferences

1. T. Hayashi, N. Kitaoka, C. Miyajima, K. Takeda, “Investigating the robustness of deep bottleneck features for recognizing speech of speakers of various ages,” Proc. FORUM ACUSTICUM, 4 pages, Sep, 2014.

2. N. Kitaoka, T. Hayashi, K. Takeda, “Noisy speech recognition using blind spatial subtraction array technique and deep bottleneck features,” Proc. APSIPA, 4 pages, Dec. 2014.

3. T. Hayashi, M. Nishida, N. Kitaoka, K. Takeda, “Daily activity recognition based on DNN using environmental sound and acceleration signals,” Proc. EUSIPCO, pp. 2351–2355, Sep. 2015.

4. H. Erdogan, T. Hayashi, J. R. Hershey, T. Hori, C. Hori, W. Hsu, S. Kim, J. LeRoux, Z. Meng, S. Watanabe, “Multi-channel speech recognition: LSTMs all the way through,” Proc. CHiME4 workshop, 2016.

5. T. Hayashi, S. Watanabe, T. Toda, T. Tori, J. LeRoux, K. Takeda, “Convolutional bidirectional long short-term memory hidden Markov model hybrid system for polyphonic sound event detection,” Proc. 5th Joint Meeting of the Acoustical Society of America and Acoustical Society of Japan, Dec. 2016.

6. S. Araki, T. Hayashi, M. Delcroix, M. Fujimoto, K. Takeda, T. Nakatani, “Exploring multi-channel features for denoising-autoencoder-based speech enhancement, ” Proc. ICASSP, pp.116–120, Apr. 2015.

7. T. Hayashi, S. Watanabe, T. Toda, T. Hori, J. Le Roux, K. Takeda, “Bidirectional LSTM-HMM hybrid system for polyphonic sound event detection,” Proc. DCASE2016 workshop, 5 pages, Sep. 2016.

8. A. Tamamori, T. Hayashi, T. Toda, K. Takeda, “Investigation on recurrent neural network architectures for daily activity recognition,” Proc. UV2016, Oct. 2016.

9. T. Hayashi, S. Watanabe, T. Toda, T. Hori, J. Le Roux, K. Takeda, “BLSTM-HMM hybrid system combined with sound activity detection network for polyphonic sound event detection,” Proc. ICASSP, pp. 766–770, Mar. 2017.

10. K. Kobayashi, T. Hayashi, A. Tamamori, T. Toda, “Statistical voice conversion with WaveNet-based waveform generation,” Proc. INTERSPEECH, pp. 1138–1142, Aug. 2017.

11. A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, T. Toda, “Speaker-dependent WaveNet vocoder,” Proc. INTERSPEECH, pp. 1118–1122, Aug. 2017.

12. A. Tamamori, T. Hayashi, T. Toda, K. Takeda, “Investigation of effectiveness on recurrent neural network for daily activity recognition using multi-modal signals,” Proc. APSIPA, pp. 1334–1340, Dec. 2017.

13. T. Hayashi, A. Tamamori, K. Kobayashi, K. Takeda, T. Toda, “An investigation of multi-speaker training for WaveNet vocoder,” Proc. ASRU, pp. 712–718, Dec. 2017.

14. P. L. Tobing, Y. Wu, T. Hayashi, K. Kobayashi, T. Toda, “NU voice conversion system for the voice conversion challenge 2018,” Proc. Odyssey 2018, pp. 219–226, June 2018.

15. Y. Wu, P. L. Tobing, T. Hayashi, K. Kobayashi, T. Toda, “The NU non-parallel voice conversion system for the voice conversion challenge 2018,” Proc. Odyssey 2018, pp. 211–218, June 2018.

16. Y. Wu, K. Kobayashi, T. Hayashi, P. L. Tobing, T. Toda, “Collapsed segment detection and reduction for WaveNet vocoder,” Proc. INTERSPEECH, pp. 1998–1992, Sep. 2018.

17. T. Hayashi, S. Watanabe, T. Toda, K. Takeda, “Multi-head decoder for end-to-end speech recognition,” Proc. INTERSPEECH, pp. 801–805, Sep. 2018.

18. T. Hayashi, T. Komatsu, R. Kondo, T. Toda, K. Takeda, “Anomalous sound event detection based on WaveNet,” Proc. EUSIPCO, pp. 2508–2512, Sep. 2018.

19. K. Miyazaki, T. Hayashi, T. Toda, K. Takeda, “Connectionist temporal classification-based sound event encoder for converting sound events into onomatopoeia representations,” Proc. EUSIPCO, pp. 857–861, Sep. 2018.

20. P. L. Tobing, T. Hayashi, Y. Wu, K. Kobayashi, T. Toda, “An evaluation of deep spectral mappings and WaveNet vocoder for voice conversion,” Proc. IEEE SLT, pp. 297–303, Dec. 2018.

21. T. Hayashi, S. Watanabe, Y. Zhang, T. Toda, T. Hori, R. Astudillo, K. Takeda, “Back-translation-style data augmentation for end-to-end ASR,” Proc. IEEE SLT, pp. 426–433, Dec. 2018.

## Review Papers

1. K. Miyazaki, T. Toda, T. Hayashi, K. Takeda, “Environmental sound processing and its applications,” IEEJ Transactions on Electronics, Information and Systems, Vol. 14, No. 3, Mar. 2019. (to appear)

## Domestic Conferences

1. T. Hayashi, N. Kitaoka, K. Takeda, “Investigating the robustness of deep bottleneck features for recognizing speech of speakers of various ages,” Proc. ASJ, pp.77–80, Sep,

2014 (in Japanese).

2. T. Hayashi, M. Nishida, N. Kitaoka, K. Takeda, “Daily activity recognition based on DNN using enviormental sound and acceleration signals,” Proc. ASJ, pp. 83–86, Mar. 2015.

3. S. Araki, T. Hayashi, M. Delcroix, M. Fujimoto, K. Takeda, T. Nakatani, “Multi-channel features for DAE-based speech enhancement,” Proc. ASJ, pp. 685–686, Mar. 2015 (in Japanese).

4. T. Hayashi, K. Ohtani, K. Takeda, “Investigation of sound quality improvement of lossy audio signals based on deep neural network,” Proc. ASJ, pp. 537–538, Sep. 2015 (in Japanese).

5. T. Hayashi, N. Kitaoka, T. Toda, K. Takeda, “Adaptation methods for daily activity recognition based on deep neural network,” IEICE Tech. Rep., Vol. 116, No. 189, SP2016–27, pp. 1–6, Aug. 2016 (in Japanese).

6. A. Tamamori, T. Hayashi, T. Toda, K. Takeda, “Daily activity recognition based on recurrent neural network ,” IEICE Tech. Rep., Vol. 116, No. 189, SP2016–28, pp. 7–12, Aug. 2016 (in Japanese).

7. A. Tamamori, T. Hayashi, T. Toda, K. Takeda, “Investigation on recurrent neural network architectures for daily activity recognition,” Proc. ASJ, pp. 1–2, Sep. 2016 (in Japanese).

8. A. Tamamori, T. Hayashi, T. Toda, K. Takeda, “Speech waveform synthesis based on WaveNet considering speech generation process,” IEICE Tech. Rep., Vol. 116, No. 477, SP2016–77, pp. 1–6, Mar. 2017 (in Japanese).

9. T. Hayashi, S. Watanabe, T. Toda, T. Hori, J. Le Roux, K. Takeda, “Explicit event duration-controlled BLSTM-HSMM hybrid model for polyphonic sound event detection ,” IEICE Tech. Rep., Vol. 117, No. 138, EA2017-2, pp. 9–14, July 2017 (in Japanese).

10. T. Hayashi, S. Watanabe, T. Toda, T. Hori, J. Le Roux, K. Takeda, “BLSTM-HMM hybrid system combined with sound activity detection network for polyphonic sound event detection,” Proc. ASJ, pp. 45–46, Mar. 2017 (in Japanese).

11. S. Noda, T. Hayashi, T. Toda, K. Takeda, “DNN acoustic model training using normal speech for non-audible murmur recognition,” Proc. ASJ, pp. 89–90, Mar. 2017 (in Japanese).

12. S. Watanabe, T. Hori, T. Hayashi, K. Suyoun, “End-to-end Japanese ASR without using morphological analyzer, pronunciation dictionary and language model,” Proc. ASJ, pp. 29–30, Mar. 2017.

13. T. Hayashi, A. Tamamori, K. Kobayashi, K. Takeda, T. Toda, “A study on the use of multi-speaker data for WaveNet vocoder training,” Proc. ASJ, pp. 285–286, Sep. 2017 (in Japanese).

14. K. Miyazaki, T. Hayashi, T. Toda, K. Takeda, “Conversion from sound event to onomatopoeia representation based on CTC,” Proc. ASJ, pp. 19–20, Sep. 2017 (in Japanese).

15. S. Noda, T. Hayashi, T. Toda, K. Takeda, “Development of speaker/environment-dependent acoustic model for non-audible murmur recognition based on DNN adaptation,” IEICE Tech. Rep., Vol. 117, No. 368, EA2017–56, pp. 7–10, Dec. 2017 (in Japanese).

16. K. Kobayashi, T. Hayashi, A. Tamamori, T. Toda, “Statistical voice conversion with WaveNet vocoder,” IEICE Tech. Rep., Vol. 117, No. 393, SP2017–82, pp. 87–92, Jan. 2018 (in Japanese).

17. T. Hayashi, K. Kobayashi, A. Tamamori, K. Takeda, T. Toda, “An investigation of multi-speaker WaveNet vocoder,” IEICE Tech. Rep., Vol. 117, No. 393, SP2017–81, pp. 81–86, Jan. 2018 (in Japanese).

18. T. Hayashi, K. Kobayashi, A. Tamamori, K. Takeda, T. Toda, “A study on an effect of the amount of training data for WaveNet vocoder,” 音講論, pp. 249–250, Mar. 2018 (in Japanese).

19. P. L. Tobing, Y. Wu, T. Hayashi, K. Kobayashi, T. Toda, “Development of NU voice conversion system for Voice Conversion Challenge 2018,” Proc. ASJ, pp. 215–216, Mar. 2018.

20. Y. Wu, P. L. Tobing, T. Hayashi, K. Kobayashi, T. Toda, “Development of NU non-parallel voice conversion system for Voice Conversion Challenge 2018,” Proc. ASJ, pp. 217–218, Mar. 2018.

21. K. Miyazaki, T. Hayashi, T. Toda, K. Takeda, “Sound event encoder using onomatopoeic representations based on end-to-end approach ,” IEICE Tech. Rep., Vol. 118, No. 198, SP2018–30, pp. 37–42, Aug. 2018 (in Japanese).

22. S. Seki, T. Hayashi, K. Takeda, T. Toda, “Waveform generation from magnitude spectrogram based on WaveNet,” Proc. ASJ, pp. 281–282, Sep. 2018 (in Japanese).

23. T. Hayashi, S. Watanabe, T. Toda, K. Takeda, “Multi-head decoder network for attention-based end-to-end ASR,” Proc. ASJ, pp. 925–926, Sep. 2018 (in Japanese).

# Awards

1. 日本音響学会第12回 学生優秀発表賞, Sep. 2014.

2. 平成27年度電子情報通信学会東海支部 学生研究奨励賞, July 2015.