creative commons

COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

공학박사학위논문

# Data-Efficient and Weakly Supervised Techniques for Audio Event Detection

## 음향 이벤트 탐지를 위한 효율적 데이터 활용 및 약한 교사학습 기법

2020년 2월

서울대학교 대학원

전기 · 컴퓨터공학부

최 인 규

# Abstract

Conventional audio event detection (AED) models are based on supervised approaches. For supervised approaches, strongly labeled data is required. However, collecting large-scale strongly labeled data of audio events is challenging due to the diversity of audio event types and labeling difficulties. In this thesis, we propose data-efficient and weakly supervised techniques for AED.

In the first approach, a data-efficient AED system is proposed. In the proposed system, data augmentation is performed to deal with the data sparsity problem and generate polyphonic event examples. An exemplar-based noise reduction algorithm is proposed for feature enhancement. For polyphonic event detection, a multi-labeled deep neural network (DNN) classifier is employed. An adaptive thresholding algorithm is applied as a post-processing method for robust event detection in noisy conditions. From the experimental results, the proposed algorithm has shown promising performance for AED on a low-resource dataset.

In the second approach, a convolutional neural network (CNN)-based audio tagging system is proposed. The proposed model consists of a local detector and a global classifier. The local detector detects local audio words that contain distinct characteristics of events, and the global classifier summarizes the information to predict audio events on the recording. From the experimental results, we have found that the proposed model outperforms conventional artificial neural network models.

In the final approach, we propose a weakly supervised AED model. The proposed model takes advantage of strengthening feature propagation from DenseNet and modeling channel-wise relationships by SENet. Also, the correlations among segments in audio recordings are represented by a recurrent neural network (RNN) and conditional random field (CRF). RNN utilizes contextual information and CRF post-processing helps to refine segment-level predictions. We evaluate our proposed method and compare its performance with a CNN based baseline approach. From a number of experiments, it has been shown that the proposed method is effective both on audio tagging and weakly supervised AED.

**Keywords:** Audio event detection, data-efficient, weakly supervised learning, deep learning.

**Student number:** 2012-23248

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Audio signals carry a large amount of information about our human activities and physical events with meaningful information. Audio event detection (AED) is a key role to utilize this information. AED aims to identify the occurrence of specific sounds in audio recordings. As the amount of multimedia data on the internet is growing rapidly, analyzing audio events will help to describe and to understand environmental and social activities in video and audio content. AED is also useful in many other applications, including surveillance, self-driving cars, healthcare, smart home systems, and military applications.

In early studies on AED, several approaches were proposed based on signal processing and machine learning techniques. Several approaches were proposed based on Gaussian mixture model (GMM) and hidden Markov model (HMM), similar to speech recognition techniques. In other studies, support vector machine (SVM) was also applied to AED as a classifier. Non-negative matrix factorization (NMF) was used to represent audio events as a combination of bases in some studies. Bag of words representation was used to represent and detect audio events with various classifiers. Recently, deep learning methods have been widely applied in AED. Deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs) were used to classify audio events.

One of the difficulties in learning AED models is the lack of fully supervised data. In conventional AED approaches, strongly labeled data is used. In strongly labeled data, either audio event examples are directly provided, or the exact time of each audio event is given. Building a large fully labeled database is a time-consuming and challenging work because it is difficult for humans to identify the onsets and offsets of audio events exactly. For this reason, there exist only a few publicly available large-scale fully supervised audio event datasets. Therefore, supervised AED models should consider the data sparsity problem when where is not enough data. Another way to deal with the lack of fully supervised data is to utilize weakly supervised datasets. In weakly supervised datasets, only the presences or absences of events in the recording are provided. Therefore, we can obtain weakly labeled datasets much easier than strongly labeled datasets.

In this thesis, we propose data-efficient and weakly supervised techniques for AED. The proposed approaches are based on deep learning methods and can be applied to different theoretical cases for audio event detection. In Chapter 3, we propose data-efficient techniques for AED. In the proposed system, data augmentation is performed to deal with the data sparsity problem in a small training dataset and generate polyphonic event examples. An exemplar-based noise reduction algorithm is proposed for feature enhancement. A DNN classifier is used for polyphonic event detection and an adaptive thresholding algorithm is applied as post-processing for robust event detection in noisy conditions. The proposed algorithm was evaluated on the DCASE 2016 task 2 dataset and showed good performance.

In Chapter 4, we propose an audio tagging system on weakly supervised data, which is labeled with only the existence of events. The model consists of a local detector and a global classifier. The local detector detects local audio words that contain distinct characteristics of events, and the global classifier summarizes the information to make a decision on the recording. The experiments show that the proposed model has better performance on the CHiME Home dataset than other neural network-based models.

In Chapter 5, we propose an AED model based on DenseNet and SENet for weakly supervised AED. We take advantage of strengthening feature propagation from DenseNet and modeling channel-wise relationships by SENet. Also, the correlations among segments in recordings are considered through an RNN and conditional random field (CRF) [1]. We evaluate our proposed method and compare its performance with a CNN based baseline approach. Empirical results show that the proposed method outperforms the baseline on the DCASE 2017 task 4 dataset.

The rest of this thesis is organized as follows: The next chapter introduces the different theoretical cases for audio event detection covered in this thesis. In Chapter 3, we propose data-efficient techniques for AED. In Chapter 4, we propose the audio tagging system for weakly supervised data, which is labeled with only the existence of events. In Chapter 5, we propose the deep CNN based on DenseNet and SENet for weakly supervised AED. The conclusions are drawn in Chapter 6.

# Chapter 2

# Audio Event Detection

There are different theoretical cases for audio event detection, depending on the information to be estimated. The term audio event classification is used to indicate a multi-class single-label case, where a single audio sample is assigned to a single event class. When multiple labels are assigned to a single audio sample, the task is referred to as audio tagging. In audio event detection, the presence and temporal activity of audio events in audio recordings are estimated. In our study, we focus on audio event detection and audio tagging. In order to evaluate the performance of the proposed techniques, the datasets suitable for each task are required. In the case of speech, there are databases commonly used in many studies such as TIMIT, Aurora-4 DB, and LibreSpeech, so it is easy to compare the results of the studies. However, AED studies often use different databases depending on the targeted acoustic events, and there are some studies that use unpublished databases. Therefore, it is important to select an appropriate database to confirm the performance of the algorithms. In order to fairly compare the proposed algorithm with other studies, we conducted the study using a DCASE challenge database that many researchers utilize. DCASE challenge was first organized in 2013 and has been held annually since 2016. The subject of the DCASE challenge is the computational auditory scene analysis (CASA), which covers AED and scene classification. The challenge aims to provide open data for researchers to use in their work and successive reference points for performance comparison for AED algorithms. The DCASE challenge consists of four to five tasks that change every year, and each task is given a database that fits its purpose. We used databases from the tasks suitable for our studies.

## 2.1 Data-Efficient Audio Event Detection

AED is defined as the task of finding individual audio events in audio recordings by indicating onset, offset, and class labels for each audio event. In general, AED models are trained by supervised methods that require strongly labeled data. In strongly labeled data, either acoustic event examples are directly provided, or the exact locations of the acoustic events in the recordings are given so that specific events can be extracted from the entire recordings. However, there are not many publicly available large-scale datasets with strong labels. Therefore, it is important that AED models should be robust even when there is not enough data.

We used the DCASE 2016 task2 dataset for low-resource supervised AED. DCASE 2016 task2 focused on event detection of office audio events and will use training material provided as isolated audio events for each class, and synthetic mixtures of the same examples in multiple SNR and event density conditions. The test data consists of synthetic mixtures of audio events at various SNR levels, event density conditions, and polyphony. The training dataset was composed of mono recordings of isolated acoustic events typically found in an office environment. 11 classes

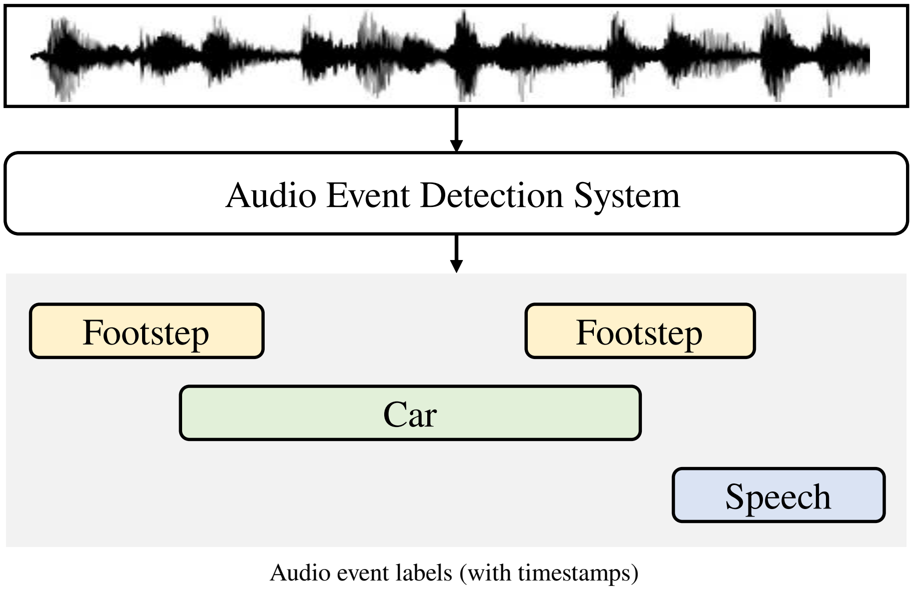
Audio Event Detection System
Footstep
Footstep
Car
Speech
Audio event labels (with timestamps)

Figure 2.1: Audio event detection system.

were available: clearthroat, cough, doorslam, drawer, keyboard, keys, knock, laughter, pageturn, phone, speech and each class was represented by 20 recordings in the training dataset. The development dataset consisted of 18 two-minute recordings in various noise and event density conditions. The test dataset consisted of 54 two-minute recordings similar to the development dataset. The training and development dataset was used for training the model, and the test dataset was used only for performance evaluation.

## 2.2 Audio Tagging

Audio tagging is defined as a multi-label classification problem, in which each possible label corresponds to a class of audio events which may occur in the audio sample. Unlike AED, the onset and offset of events are not requested in audio tagging. Thus, weakly labeled data without timestamps of audio events can be used

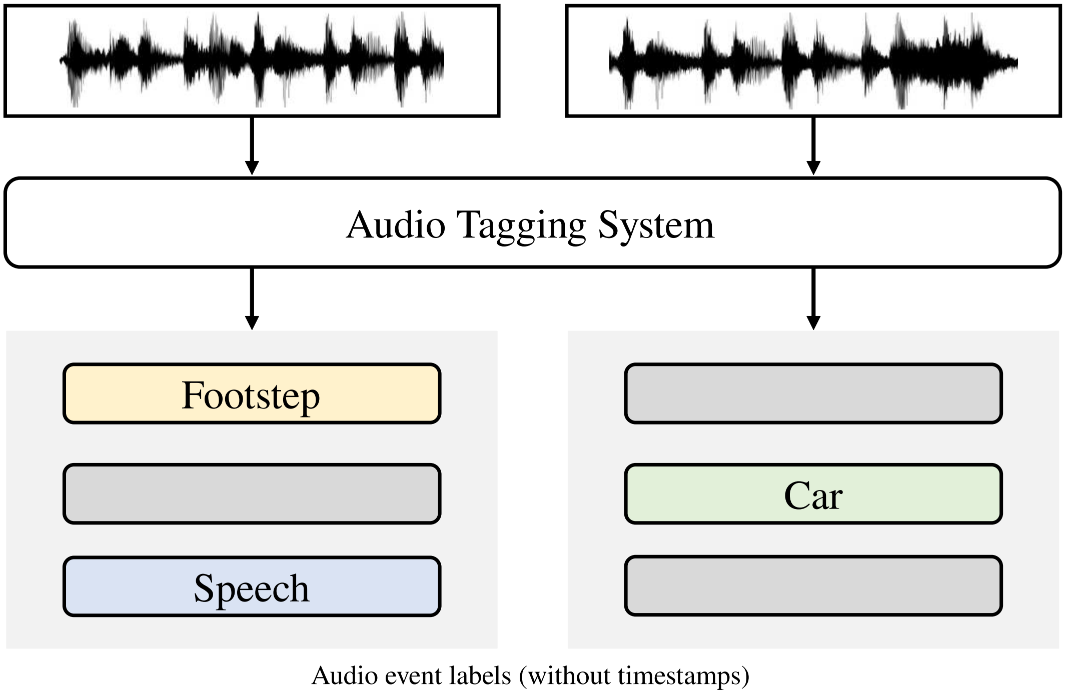

Figure 2.2: Audio tagging system.

for learning an audio tagging system. A practical benefit of audio tagging comes from the uncomplicated annotation process, which does not require manual event boundaries.

For the evaluation of the proposed audio tagging algorithm, we employ the CHiME-Home dataset [2], which was used in DCASE 2016 task 4. The acoustic environment comprises the following audio sources: Two adults and two children, television and electronic gadgets, kitchen appliances, footsteps, and knocks produced by human activity, further to sound originating from outside the house. The audio data are provided at sampling rates of 16 kHz single-channel recordings. The development set consists of 4378 recordings, and another 1759 recordings are used for evaluation. Each recording has 4 seconds duration. The event classes in the dataset are child speech, male speech, female speech, TV, percussive sounds, broadband noise, and other identifiable sounds. The dataset includes only record-level labels without timestamps.

## 2.3 Weakly Supervised Audio Event Detection

In early studies on AED, most of the studies were based on fully supervised learning methods that require strongly labeled data. In strongly labeled data, either audio event examples are directly provided, or the exact time of each audio event is given. However, building a large strongly labeled database is a time-consuming and challenging work. For these reasons, there exist only a few publicly available large-scale audio event datasets with strong labels. In weakly supervised approaches, AED models are learned based on weakly labeled data that provides only the presence or absence of events in the recording. The difference with audio tagging is that weakly supervised AED also aims to predict the onset and offset in addition to the presence of events. The task thus raises an interesting technical challenge, how to learn a model that predicts strong labels with timestamps from weakly labeled data without timestamps.

We used the DCASE 2017 task4 dataset for the evaluation of the proposed algorithm. The DCASE 2017 task 4 Dataset [3] was published for the task of "Large-scale weakly supervised sound event detection for smart cars" in the DCASE 2017 challenge. The dataset employs a subset of AudioSet by Google [4]. The DCASE 2017 task 4 Dataset consists of 17 audio events divided into two categories: "Warning" and "Vehicle". The dataset contains audio classes for self-driving cars, smart cities, and related areas. The dataset contains 51,172 clips of the training set, 488 clips of the

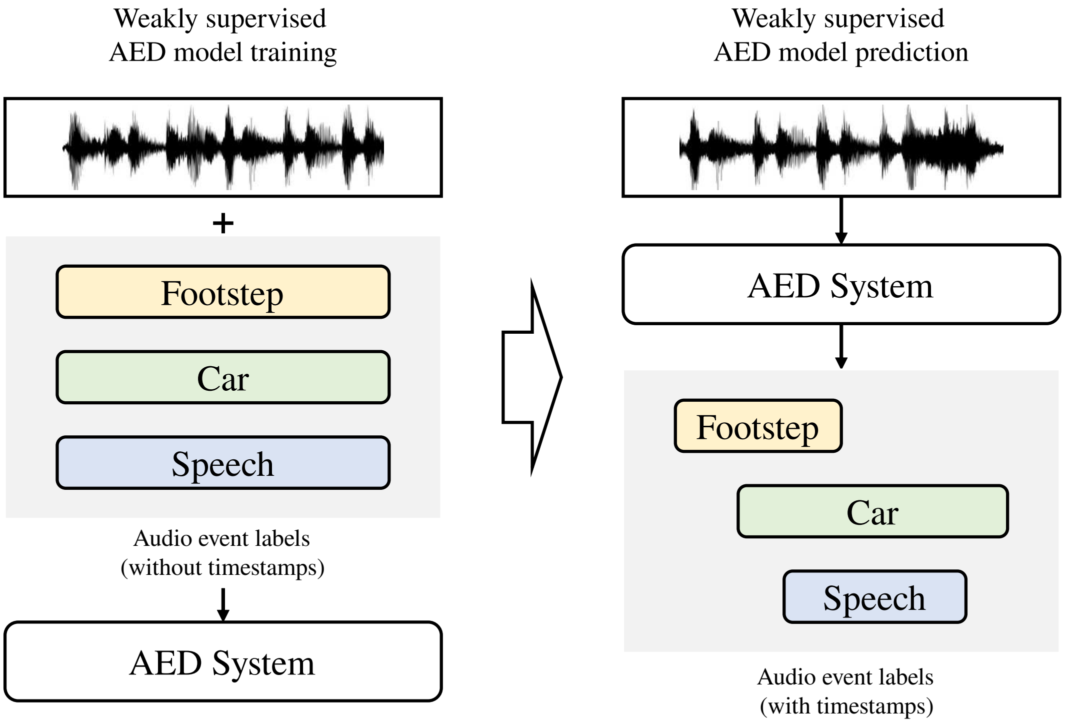

Figure 2.3: Weakly supervised audio event detection system.

validation set, and 1,103 clips of evaluation set. Every clip is less than 10 seconds long. Each clip may correspond to more than one audio event and possibly has overlapping audio events. The dataset is obtained by collecting real-life recordings that contain noise and unknown class signals. The training set has weak labels denoting the presence of a given audio event in the clip, and no timestamps are provided. For the validation and evaluation sets, strong labels with timestamps are provided for the purpose of performance evaluation.

## 2.4 Metrics

Evaluation of the performance of AED systems is done by comparing the system output with a reference available for the test data. Suitable metrics are required for

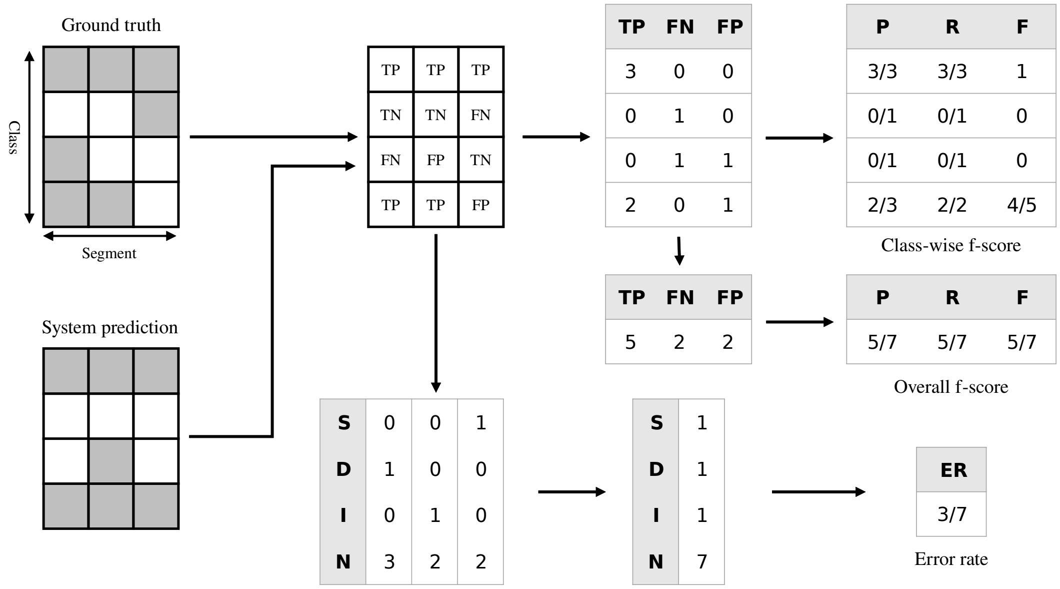

Figure 2.4: Calculation of f-score and error rate.

the performance comparison of the AED systems. Metrics from neighboring fields such as speech recognition can be used, but they need to be partially redefined to deal with multi-label classification results. In our study, we used the metrics for measuring the performance of polyphonic AED proposed in [5].

In AED tasks, the comparison between the system output and reference can be done in fixed-length intervals. These intervals may be short or the entire length of the audio signal. For evaluation of metrics, we need to define what constitutes correct detection and what type of errors the system produces. These are referred to as intermediate statistics that count the correct and incorrect predictions of the system separately. For intermediate statistics, the active/inactive state of each event class is determined in a fixed-length interval. The intermediate statistics are defined as:

- true positive (TP): the reference and system prediction both indicate an event to be active in the segment.

- true negative (TN): the reference and system prediction both indicate an event to be inactive in the segment.
- false positive (FP): the reference indicates an event to be inactive in the segment, but the system prediction indicates it as active.
- false negative (FN): the reference indicates an event to be active in the segment, but the system prediction indicates it as inactive.

Base on these intermediate statistics, precision (P), recall (R), and f-score (F) are introduced. Precision is the fraction of correctly retrieved instances among all retrieved instances, and recall is the fraction of correctly retrieved instances among all relevant instances. They are defined as

$$P = \frac{TP}{TP + FP} \tag{2.1}$$

$$R = \frac{TP}{TP + FN}. \tag{2.2}$$

Based on precision and recall, f-score is determined as a measure of effectiveness of retrieval. F-score is calculated as

$$F = \frac{2 \cdot P \cdot R}{P + R} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}. \tag{2.3}$$

Error rate (ER) represents the number of errors in terms of insertions (I), deletions (D), and substitutions (S). ER is calculated in a segment-wise manner. In a segment $k$, the number of substitution errors $S(k)$ is the number of predicted false events when the reference indicates other undetected events. This is determined by pairing false positives and false negatives, without considering which erroneous event substitutes which. The number of deletion errors $D(k)$ is the number of missing events that the reference indicates active events, but the system doesn't detect (false negatives after substitutions are accounted for). The number of insertion errors $I(k)$ is the number of predicted false events that the reference indicates no events (false negatives after substitutions are accounted for). They are defined as

$$S(k) = min(FP(k), FN(k)) \tag{2.4}$$

$$D(k) = max(0, FN(k) - FP(k)) \tag{2.5}$$

$$I(k) = max(0, FP(k) - FN(k)). \tag{2.6}$$

Total error rate is calculated by integrating segment-wise counts over the total number of segments, given as

$$ER = \frac{\sum S(k) + \sum D(k) + \sum I(k)}{\sum N(k)} \tag{2.7}$$

where N(k) is the number of active events at segment $k$ in the reference. The calculation of f-score and error rate is illustrated in Fig. 2.4.

Equal error rate (EER) is also used for performance evaluation. To calculate EER, FP and FN errors are used. Multi-label AED systems generally output scores for each event. This score is a scalar variable that represents the possibility of an event to be present. To make a decision, the AED system needs to use a threshold value to decide whether an event is active or not. If the threshold is too low, there will be a lot of FP errors, whereas if the threshold is too high, there will be too

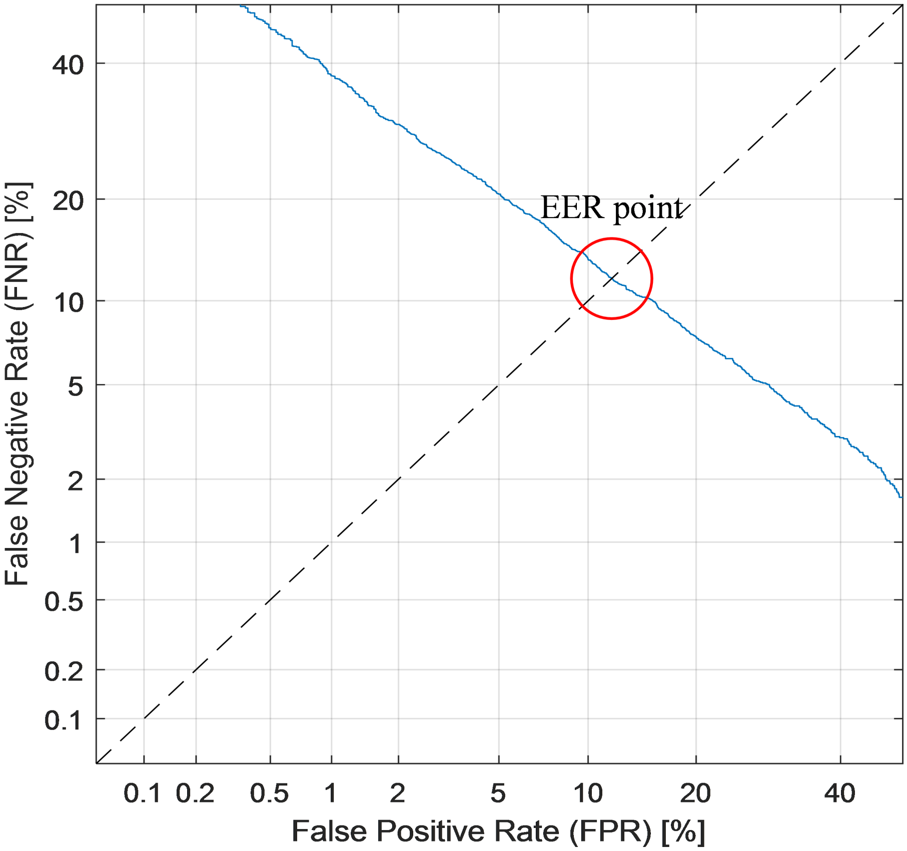

Figure 2.5: Example of a DET curve.

many FN errors. According to these definitions, two error rates are defined as

$$False\,Positive\,Rate\,(FPR) = \frac{Number\,of\,FP\,errors}{Number\,of\,inactive\,events\,in\,the\,reference} \tag{2.8}$$

$$False\,Negative\,Rate\,(FNR) = \frac{Number\,of\,FN\,errors}{Number\,of\,active\,events\,in\,the\,reference}. \tag{2.9}$$

The EER is defined as the FPR and FNR values when they become equal. To calculate EER, we should find a point where the FAR and FRR become equal by changing the threshold. To represent the performance of detection systems graphically, the detection error tradeoff (DET) curve is generally employed. The DET curve is a plot of the FPR versus FNR on different thresholds. An example DET curve is shown in Fig. 2.5. When the performance of the system is better, the curve is closer to the origin.

# Chapter 3

# Data-Efficient Techniques for Audio Event Detection

## 3.1 Introduction

Audio event detection plays an important role in computational auditory scene analysis, with a specific purpose of detecting meaningful sounds, generally referred to as audio events. Detecting audio events such as speech, footstep, and door slam provides fundamental information for understanding the situation using acoustic signals. Furthermore, AED could be utilized in many applications, including automated surveillance systems, information retrieval, smart home systems, and military applications.

Many previous works on AED were based on conventional speech recognition techniques. The most common approach is to use a system based on spectral features such as mel-frequency cepstral coefficients (MFCCs) and HMM for audio event classification [6], [7]. In recent works, approaches based on SVM [8]–[10] or NMF [11]–

[13] were also proposed for AED. Most of the previous works were monophonic AED, which focused on detecting a single event at the same time. However, more than two events can happen simultaneously in real environments. In this case, conventional monophonic AED approaches may not be suitable for detecting overlapping events. Polyphonic AED aims to detect multiple audio events in the same time instance of the audio data. A polyphonic AED system that used MFCC for feature and HMMs as classifiers with consecutive passes of the Viterbi algorithm was proposed [14]. In [15], a generalized Hough transform (GHT) voting system has been used to recognize overlapping audio events. In another work, the NMF-based approach was used for source separation, and then events were detected from each stream [16]. In [17], they showed that DNNs outperform the SVM and GMM models. DNNs have shown good performance for polyphonic AED by modeling overlapping audio events in a natural way [18].

In this chapter, we propose a DNN-based data-efficient AED system. In the proposed system, data augmentation is performed to deal with the data sparsity problem in small training datasets and generate polyphonic event examples. An exemplar-based noise reduction algorithm is proposed for feature enhancement. DNN classifier is trained for polyphonic event detection, and an adaptive thresholding algorithm is applied as post-processing for robust event detection in noisy conditions.

## 3.2 DNN-Based AED system

The proposed system consists of 4 main processing stages. The overall system is illustrated in Fig. 3.1. First, data augmentation is performed to generate artificial audio event scenes that are used for training the classifier. In the second stage, mel

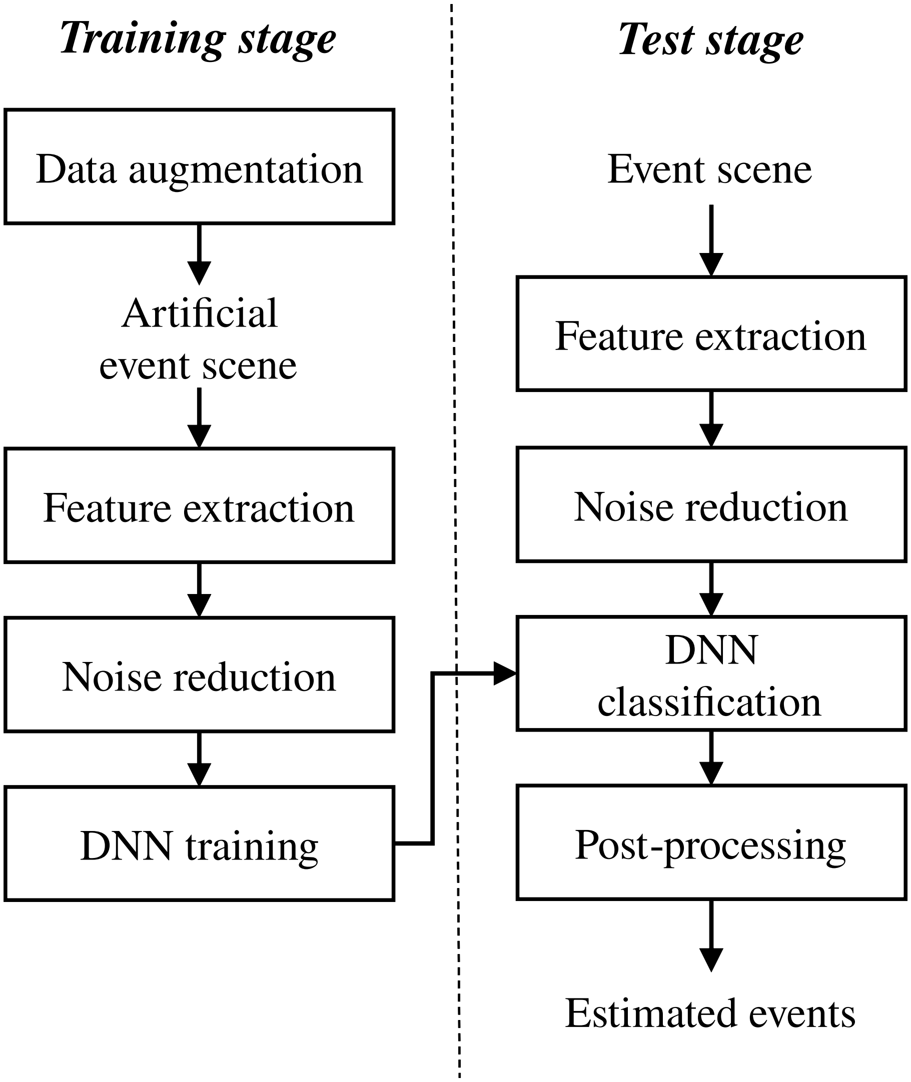

Figure 3.1: Flowchart of the proposed system.

filterbank features are extracted and enhanced by exemplar-based noise reduction. Third, the enhanced feature is fed to a DNN classifier. The features from artificial audio event scenes are used for training the DNN classifier. In the final stage, the audio events are detected by filtering and thresholding the output of the DNN classifier.

### 3.2.1 Data Augmentation

DNNs have shown good performance as classifiers in many applications. When the training data is large, the DNN could learn from the variations presented in the training data under the same labels and make classifications that are robust to intra-class variations. However, if the training data from each class is not sufficient to cover its intra-class variations, the DNN classifier trained with the data may have poor generalization ability, leading to low classification performance for test samples. In [19], the data augmentation approach was used for training DNNs to deal with the data sparsity problem.

Unlike speech datasets, which usually consist of hours of data or more, conventional audio event datasets are not sufficiently long enough to train a robust DNN classifier. Under this condition, data augmentation can help to enhance the performance of the DNN classifier by improving the generalization ability of the neural network. In recent research, data augmentation approaches were performed for better performance in polyphonic AED [20]. In our model, artificial event scenes are generated using data augmentation. In the artificial event scenes, events are overlapped with each other or manipulated by time stretching and power modification for the diversity of data. These event scenes are corrupted by white, blue, and pink noises.

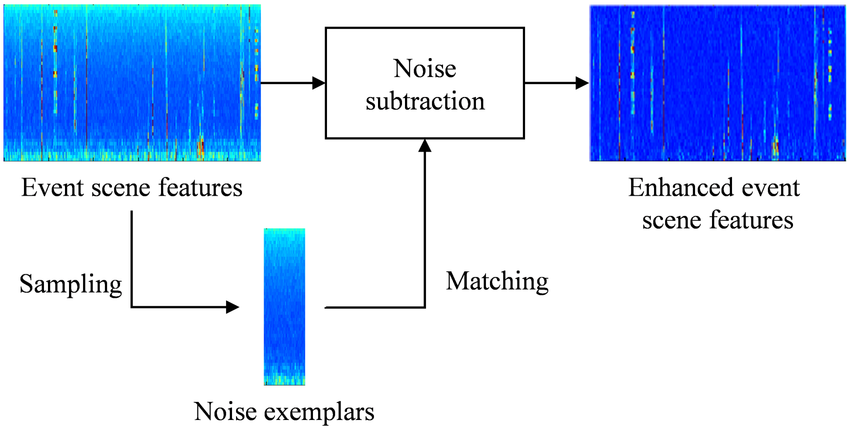

Figure 3.2: Exemplar-based approach for noise reduction.

### 3.2.2 Exemplar-Based Approach for Noise Reduction

In real life recordings, various noises exist and make it challenging to detect audio events correctly. To alleviate the effect of the noises, noise reduction is performed for feature enhancement. Since we assume that the test noise conditions are unknown, model adaptation-based approaches for noise robustness may not be suitable. In order to suppress unseen noises in test conditions, an exemplar-based noise reduction approach is proposed. In this approach, noise exemplars are selected from the event scene features, and then the noise is directly subtracted from the event scene features by using the noise exemplars.

For each event scene, mel filterbank features are extracted, and the features that have L1 norm corresponding to the lower 30% are considered to be noise candidates. From the candidates, $K$ noise frames are selected randomly or using K-means algorithm for noise exemplars. For each frame, the best matching noise exemplar that minimizes the noise estimation error, defined as in (3.1), is selected.

$$E_k = ||max(X_t - N_k, 0)||_1 + \alpha \cdot ||max(N_k - X_t, 0)||_1 \ . \tag{3.1}$$

$E_k$ is the noise estimation error of a noise exemplar $N_k$ and $X_t$ is a feature vector at time index $t$. Noise estimation error $E_k$ is the summation of underestimation error and overestimation error with a ratio of $\alpha$. The selected noise exemplar is subtracted from the frame feature for noise reduction. The proposed noise reduction process is illustrated in Fig. 3.2.

### 3.2.3 DNN Classifier

We propose a DNN-based classifier for AED. Unlike speech, audio events come from different physical sources, so they possess unique characteristics that are distinct from one another. The DNN structure is employed to represent distinct audio events in a single model successfully. The DNN system for AED is illustrated in Fig. 3.3. The DNN consists of an input layer, a few hidden layers, and an output layer which are fully connected to their adjacent layers. As for the input, mel filterbank features enhanced by the proposed noise reduction approach are used. To consider temporal information, several adjacent frame features are concatenated for a single frame input. The output of the DNN is the estimated labels for input frames. The number of the output unit is the same as that of the event classes, and each output unit is matched to each class. When the event exists in the input frame, the output unit of the class is set to 1; otherwise, it is set to 0. We used rectified linear units (ReLUs) for hidden layers and sigmoid function units for the output layer.

Artificial event data generated by data augmentation is used for training the

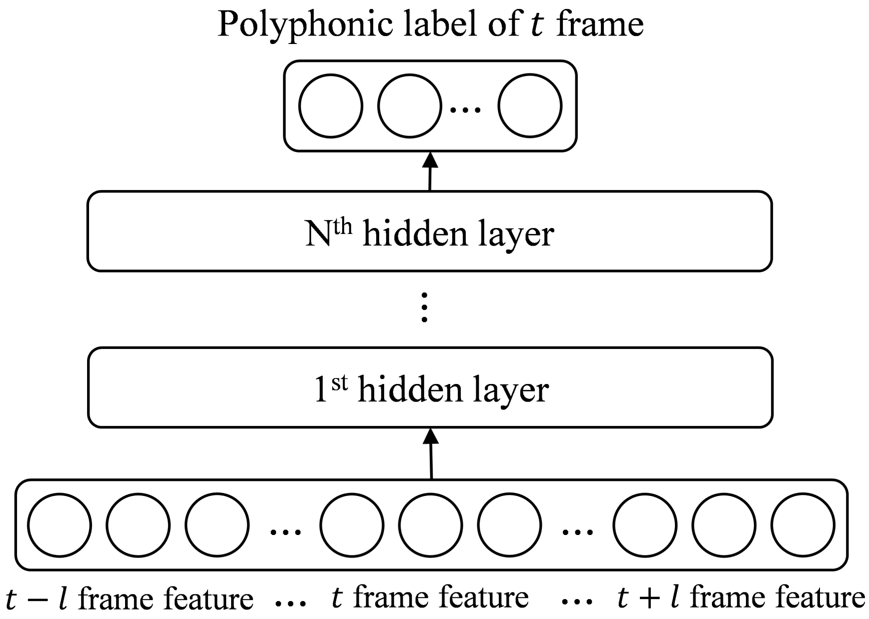

Figure 3.3: A DNN structure for the proposed AED system.

DNN classifier. In the fine-tuning stage, the backpropagation algorithm with the minimum mean squared error (MMSE) function between the correct label and the estimated label is employed to train to the DNN. A stochastic gradient descent algorithm is performed in mini-batches to improve learning convergence. To deal with the overfitting problem, we used the dropout technique, which has already proved its regularization capability for training DNN [21].

### 3.2.4 Post-Processing

The output of the DNN classifier is filtered for robust event detection. An averaging filter may help to remove outliers, but also discourage precise detection in onset or offset period of an event due to non-event periods nearby. For precise onset and offset detection, we used two filters: one of which is a sigmoid function, and the other is the former reflected about the y-axis. The former one is sensitive to the onset, and the latter one is sensitive to the offset of an event. To detect both onset and offset of an event correctly, larger values of the output of two filters are taken from both outputs of the filters.

Generally, static threshold values are used for detection. However, in noisy event scenes, static threshold values can lead to a high false detection error rate when the noise has a similar characteristic to the events. To consider the noise effect on detection, adaptive threshold values are defined as

$$th_k = th_{base,k} + \beta \cdot \frac{1}{T} \sum y_{t,k} \tag{3.2}$$

where $th_k$ is an adaptive threshold value for class $k$, $th_{base,k}$ is a base threshold value, $y_{t,k}$ is the DNN output of the class $k$ at $t$th segment. When noise characteristic is similar to class $k$, $th_k$ gets higher and reduces false detection error rate of class $k$.

## 3.3 Experiments

In order to evaluate the performance of the proposed system, we conducted experiments on the DCASE 2016 challenge task 2 dataset [22]. The training dataset was composed of mono recordings of isolated acoustic events typically found in an office environment. 11 classes were available: clearthroat, cough, doorslam, drawer, keyboard, keys, knock, laughter, pageturn, phone, speech, and each class was represented by 20 recordings in the training dataset. The development dataset consisted of 18 two-minute recordings in various noise and event density conditions. Only training dataset and noises sampled from probability density functions were used for training the system, and the development dataset was used for evaluation.

Data augmentation was performed for generating the training event scene. Each audio event scene was about two-minute-long. All events in the training dataset were normalized to have the same power, and 30 of them were randomly selected for one event scene. To diversify the training data, half of the events were manipulated by stretching the time at a ±10% rate and modifying the power in the range of 50% $\sim$ 200%. One-third of the events were overlapped to each other for polyphonic event examples. To consider the effect of noise on events, white Gaussian noise at a signal-to-noise ratio (SNR) levels 6 to 18 dB and pink noise and blue noise at SNR level 12 dB were mixed. 180 artificial event scenes were generated for training the system.

We used mel filterbank features for our system. Instead of original frequency 44.1 kHz, we used the sampling frequency of 30 kHz, spanning 50 bands between 16 Hz and 15 kHz. We used a hamming window with a frame length of 25 ms and a shift of 10 ms for frame segmentation. For noise reduction, 100 noise exemplars are selected, and $\alpha$ is set to 3. As training data and test data may have a power mismatch, the features extracted from each event scene are normalized.

For training the DNN-based classifier, 50-dimensional mel filterbank features were used as input. The input layer for DNN was formed by applying a context window of 15 frames, having 750 visible units for the network. The DNN had three hidden layers with 768 ReLUs, and the final sigmoid output layer had 11 units, each corresponding to the event classes. The fine-tuning of the network was performed using mean squared error as the loss function by error backpropagation supervised by the correct label of frames. The mini-batch size for the stochastic gradient descent algorithm was set to be 128. The dropout percentage of 20% was applied for regularization.

Table 3.1: Average detection results on the IEEE DCASE 2016 challenge task 2 evaluation dataset

| Metrics | Segment-based | Event-based |
|---|---|---|
| Precision | 0.8929 | 0.7270 |
| Recall | 0.7643 | 0.6622 |
| F-score | 0.8236 | 0.6931 |
| Substitutions | 0.0328 | 0.0337 |
| Deletions | 0.2029 | 0.3042 |
| Insertions | 0.0589 | 0.2149 |
| ER | 0.2946 | 0.5527 |

In the post-processing stage, two 11-tap sigmoid shape filters are applied for smoothing the output of the DNN. Larger values are taken from both the outputs of the filters and thresholded for event detection. We set $th_{base,k}$ to 0.8 for doorslam class and 0.6 for other classes and $\beta$ to 0.5 for adaptive thresholding. The same events within 300 $ms$ gap are concatenated, and events shorter than 150 $ms$ are removed. As evaluation measures, f-score and ER are used on the segment-based and event-based level. These measures are explained in Chapter 2.

For DCASE 2016 task 2 challenge evaluation, both training data and development data are used for training DNN classifier. The results on the evaluation dataset are shown in Table 3.1. F-score and ER on segment-based metrics are 0.8236 and 0.2946, respectively. On event-based overall metrics, F-score and ER are 0.6931 and 0.5527, respectively. Class-wise detection performance on the evaluation dataset is shown in Table 3.2. In most cases, similar performance is achieved, but it is particularly poor on the ”door slam” events. This is because the door slam event consists of concise signals, and the proposed system is not optimized for short signal detection in the post-processing stage.

Table 3.2: Class-wise detection results on IEEE DCASE 2016 Challenge Task 2 Evaluation Dataset

| Class | F-score | ER |
|---|---|---|
| Clearing throat | 0.8116 | 0.3373 |
| Coughing | 0.8090 | 0.4175 |
| Door knock | 0.8783 | 0.2268 |
| Door slam | 0.3485 | 0.8916 |
| Drawer | 0.7253 | 0.4488 |
| Keyboard | 0.9338 | 0.1379 |
| Keys | 0.8420 | 0.2920 |
| Human laughter | 0.7719 | 0.4196 |
| Page turning | 0.8590 | 0.2722 |
| Phone ringing | 0.9314 | 0.1316 |
| Speech | 0.8585 | 0.2988 |

## 3.4 Summary

We presented a data-efficient AED system based on a DNN. We used data augmentation to deal with data sparsity problem and exemplar-based approach for noise reduction. We trained a DNN for classification, and adaptive thresholding are used for detecting events. The proposed system has shown promising results on IEEE DCASE 2016 Challenge Task 2 Datasets.

# Chapter 4

# Audio Tagging using Local Detector and Global Classifier

## 4.1 Introduction

The topic of AED covers the detection and classification of acoustic events in recordings. Acoustic events help to describe and to understand the environment and social activities using acoustic information. For example, detecting acoustic events such as speech, footstep, and keyboard typing provides information on the office environment. AED could be utilized in many applications, including automated surveillance systems, information retrieval, smart home systems, and military applications.

In the early works, most of the studies focused on detecting monophonic acoustic events. Several approaches have been proposed based on HMM [6], [7], SVM [8]–[10] and NMF [11]–[13]. In real environments, more than two events can occur simultaneously. Recently, several studies have focused on polyphonic AED, which aims to detect overlapping acoustic events simultaneously in the recordings. A polyphonic

AED system that uses HMMs as classifiers with consecutive passes of the Viterbi algorithm is proposed [14]. In [16], NMF is used for source separation, and then events are detected from each stream. DNNs have shown good performance for polyphonic AED by modeling overlapping acoustic events in a natural way [18].

In previous works, most studies were based on fully supervised methods using strongly labeled data. In strongly labeled data, either acoustic event examples are directly provided, or the exact locations of the acoustic events in the recordings are given so that specific events can be extracted from the entire recordings. However, producing a large amount of strongly labeled database is a difficult and time-consuming process. Additionally, finding the exact time stamps of acoustic events is very difficult when different events occur at the same time. For these reasons, there are not many publicly available large-scale datasets with strong labels.

Recently, there have been some studies on audio tagging [23], [24]. These studies focus on learning acoustic event detectors based on weakly labeled data. In weakly labeled data, only the presence or absence of an event in the recording is known. The onset and offset of events are not requested in audio tagging. We can obtain weakly labeled datasets much easier than fully supervised datasets. For example, we can collect a huge amount of data by collecting videos uploaded on the internet. The tags of the videos can be used as weak labels for each video. However, there are some problems with using weakly labeled data for AED. The exact occurrence times of the events are not known, so learning a model for a specific class is difficult due to the interference of other data. Furthermore, It is more challenging to extract correct information when events in the recordings are overlapped.

There have been several studies on audio tagging. Kumar et al. proposed an approach of multiple instance learning (MIL) [23]. They propose two MIL frameworks based on neural networks and SVM for weakly labeled datasets. In [24], they also propose a more general framework for supervised and weakly supervised learning (SWSL). Their proposed framework can be applied for both fully supervised and weakly supervised cases. Also, various approaches were proposed for audio tagging in DCASE 2016 task 4 [25]. Several algorithms using neural network techniques such as DNNs, CNNs, and RNNs with various input features, e.g., spectrogram, MFCCs, mel-band energy, and constant Q transform (CQT) are proposed [26]–[28]. In [29], a fully convolutional neural network is applied to the UrbanSound dataset. Moreover, an event-specific Gaussian filter layer is designed to advance its learning ability. In [30], a joint detection-classification (JDC) model is proposed to detect and classify the audio recording simultaneously. The JDC model has the ability to attend to informative sounds while ignoring uninformative sounds.

In this chapter, we propose an acoustic event detection framework for weakly supervised data, which is labeled with only the existence of events. The model consists of a local detector and a global classifier. The local detector detects local audio words that contain distinct characteristics of events, and the global classifier summarizes the information to make a decision on the recording.

## 4.2 CNN-Based Audio Tagging Model

In audio tagging, we only have one multi-label for each recording. Generally, the recordings are divided into frames, so we should handle multiple frame instances with one label. One of the basic approaches is an average model. In the average model, estimated labels of frames are averaged into a record-level label. The model is trained by minimizing the cross-entropy between the averaged label and the record-

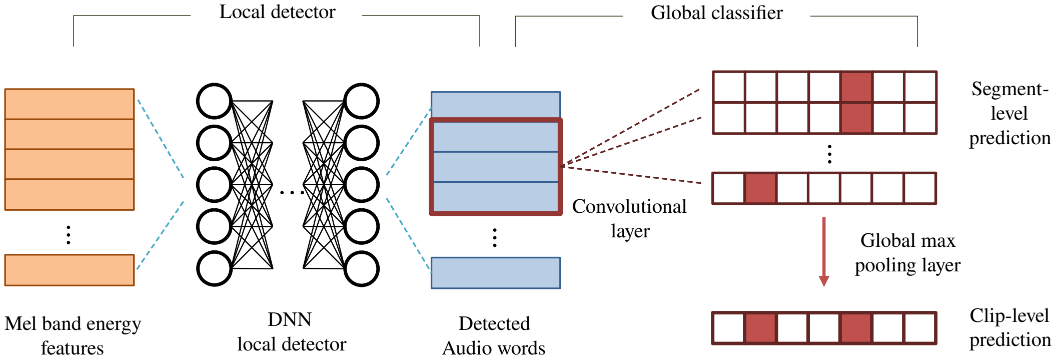

Figure 4.1: The overview of the proposed local detector and global classifier model.

level label. Thus the model uses the average label of all frames, which may cause confusion of information from different events and noise.

### 4.2.1 Local Detector and Global Classifier

In audio tagging, it is important to extract class-wise information from the recordings. One possible approach for this is to locally detect distinct characteristics of events and globally summarize the information. To do this, we propose a framework based on a local detector and global classifier (LDGC).

For the detailed analysis of the events, we can assume that each event consists of audio words. Audio words are short and distinct acoustic components of the events. In a case of voices, general phonemes, laughter sounds, and screams can be considered as audio words of voices. Moreover, different events can share audio words. For example, TV sound can have the same audio words from voice or music. These audio words can be used for describing local characteristics of each event in detail.

By detecting audio words, we can extract local characteristics from the recordings as audio words containing short-term information of the events. We use a local detector to detect audio words from each frame. The input of the local detector is a frame-level feature vector, and the output is multi-label of audio words as different audio words can occur simultaneously. Since audio words are not clearly defined, the audio words are determined by jointly training the local detector with a global classifier. The number of total audio words is determined experimentally.

The global classifier determines whether or not an event exists in the recordings based on the information of the local audio words. We use a convolutional layer with large filters to summarize the information in a large context window. The convolutional layer learns temporal correlations of the estimated frame-level audio words and finds robust estimations of the event labels. The global classifier should find an event even if there is an event only in a short interval, so we use a global max-pooling layer for the final decision.

In detail, our proposed model is shown in Fig. 4.1. For input features, we use mel filterbank features, and adjacent frame features are concatenated for temporal information. In this model, a fully connected DNN is used for the local detector. The purpose of the DNN local detector is to detect frame-level audio words. The DNN local detector takes the mel filterbank features as input and returns the estimated frame-level audio word labels. The activation function of the output layer is a sigmoid so that the detector can detect multiple audio words simultaneously. The number of the output node is $K$, which is the number of audio words. Since we can not directly train this detector as we don't have a ground truth label of audio words in each frame, the local detector should be trained jointly with the global classifier. For a global classifier, a convolutional layer and global max-pooling are used. The convolutional layer with $N$ channels is learned, where $N$ is the number of total event classes. The output from each channel determines the estimated labels for each event. The record-level label is generated by a global max-pooling layer after the convolutional layer. The global max-pooling layer selects the highest estimation score in each output channel of the convolutional layer as the final score of each class.

### 4.2.2 Temporal Localization of Events

In record-level event detection, only the existence of events is estimated. In practice, finding the locations of events in the recording is also important. The proposed model can be learned with record-level labels, but once the learning is complete, it can classify frame-level instances as the output of the convolutional layer represents the estimations of frame-level event probability.

## 4.3 Experiments

To evaluate the performance of the proposed framework, we train and evaluate three models, the baseline average model, BP-MIL [23], and the proposed LDGC model.

### 4.3.1 Dataset and Feature

For evaluation, we employ CHiME home dataset [2]. The development set consists of 4378 recordings, and another 1759 recordings are used for evaluation. Each recording has 4 seconds duration. The event classes in the dataset are “child speech”, “male speech”, “female speech”, “TV’, “percussive sounds”, “broadband noise” and “other identifiable sounds”, denoted by ‘c’, ‘m’, ‘f’, ‘v’, ‘p’, ‘b’ and ‘o’, respectively. The dataset includes only record-level labels without event positions. We used mel filterbank features as input features. Instead of the original frequency 48 kHz, we used the sampling frequency of 16 kHz, spanning 60 bands between 16 Hz and 8 kHz. We used a 30 ms Hamming windows with a 33 % overlap for frame segmentation. For temporal information, five frame features are concatenated into total 300-dimensional features.

### 4.3.2 Model Training

Three models are implemented with TensorFlow [31]. For our proposed LDGC model, a fully connected DNN is used to model the local detector. The DNN local detector consists of 3 fully connected layers with 512 units per layer. ReLU is applied to each layer, and the sigmoid function is applied to the output layer. For the number of the output node, which is the number of audio words, $K = 50$ is selected for the best performance. For the global classifier, a convolutional layer and global max-pooling are applied. The convolutional layer with filter size $50 \times 21 \times 7$ is learned. The record-level label is generated by a global max-pooling layer after the convolutional layer.

For the baseline system, a fully connected neural network similar to the local detector is used to model the frame-level classifier. The baseline model consists of 3 fully connected layers with 512 units per layer. ReLU activation function is applied to each layer. The sigmoid function is applied to the classifier's output layer, and the outputs of the classifier are averaged for record-level classification. BP-MIL has the same structure as the baseline system. However, instead of using the averaged output of the classifier for record-level classification, the maximum output of the classifier is selected for record-level classification.

We use the record-level cross-entropy error as the loss function. We apply Adam

Table 4.1: EER on CHiME home evaluation dataset.

| | Baseline | BP-MIL | LDGC |
|---|---|---|---|
| EER | 15.42% | 13.13% | 11.79% |

Table 4.2: The class-wise EER on CHiME home evaluation dataset.

| Class | BASE | MIL | LDGC |
|---|---|---|---|
| Child speech | 18.17% | 12.34% | 11.93% |
| Adult male speech | 18.37% | 12.23% | 12.61% |
| Adult female speech | 21.43% | 13.79% | 14.20% |
| Video game / TV | 8.41% | 6.73% | 6.12% |
| Percussive sounds | 25.08% | 21.03% | 19.82% |
| Broadband noise | 7.27% | 21.13% | 7.27% |
| Others | 29.60% | 28.57% | 27.41% |

[32] as our update function, and the learning rate was set to 0.00005. We set the batch size to 1, which means one recording was used for each batch. The training is stopped after 150 epochs. The dropout rate of 0.2 is applied for regularization. We use EER as our performance measure, which is explained in Chapter 2.

### 4.3.3 Results

The model belongs to the epoch with the lowest evaluation EER is selected as the best model for each algorithm. The DET curves of the models are presented in Fig. 4.2, and the EERs of the models on the evaluation set are shown in Table 4.3.2. The baseline model and BP-MIL attains EER of 15.42% and 13.13%, respectively. The proposed LDGC model has shown better performance at EER of 11.79%. The class-wise EERs of the models are shown in Table 4.2. The MIL and LDGC model have shown better performance in most classes. In most classes, the MIL and LDGC models have shown better performance than the baseline model, but the MIL model has lower performance on percussive sounds.

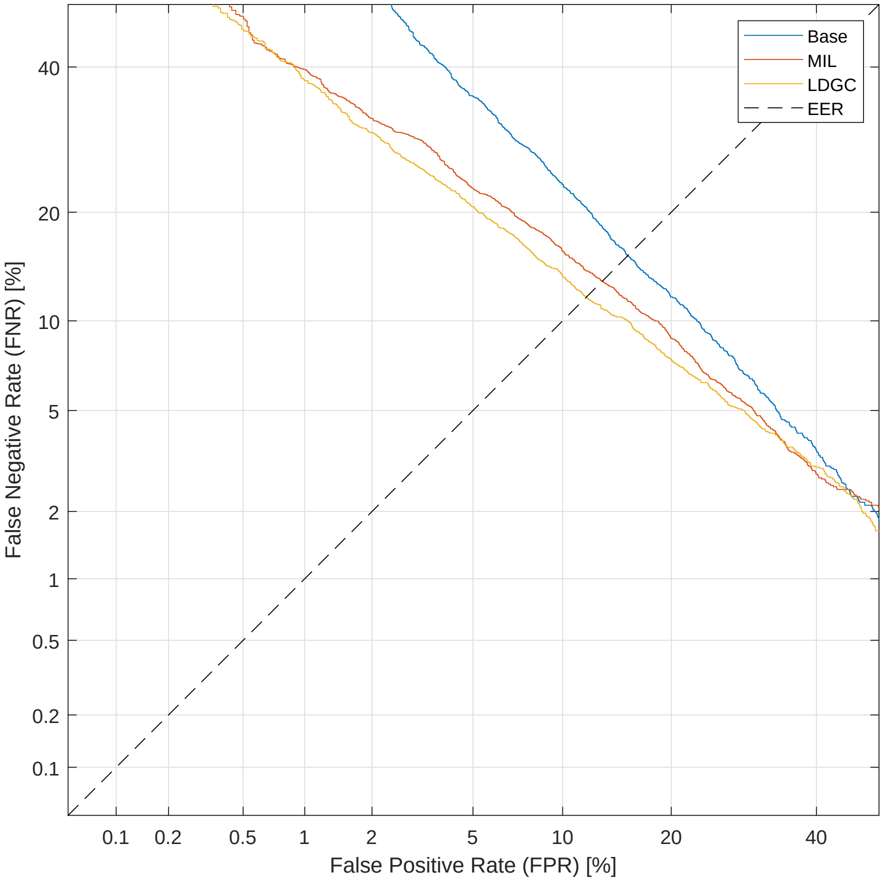

Figure 4.2: The DET curves of the models.

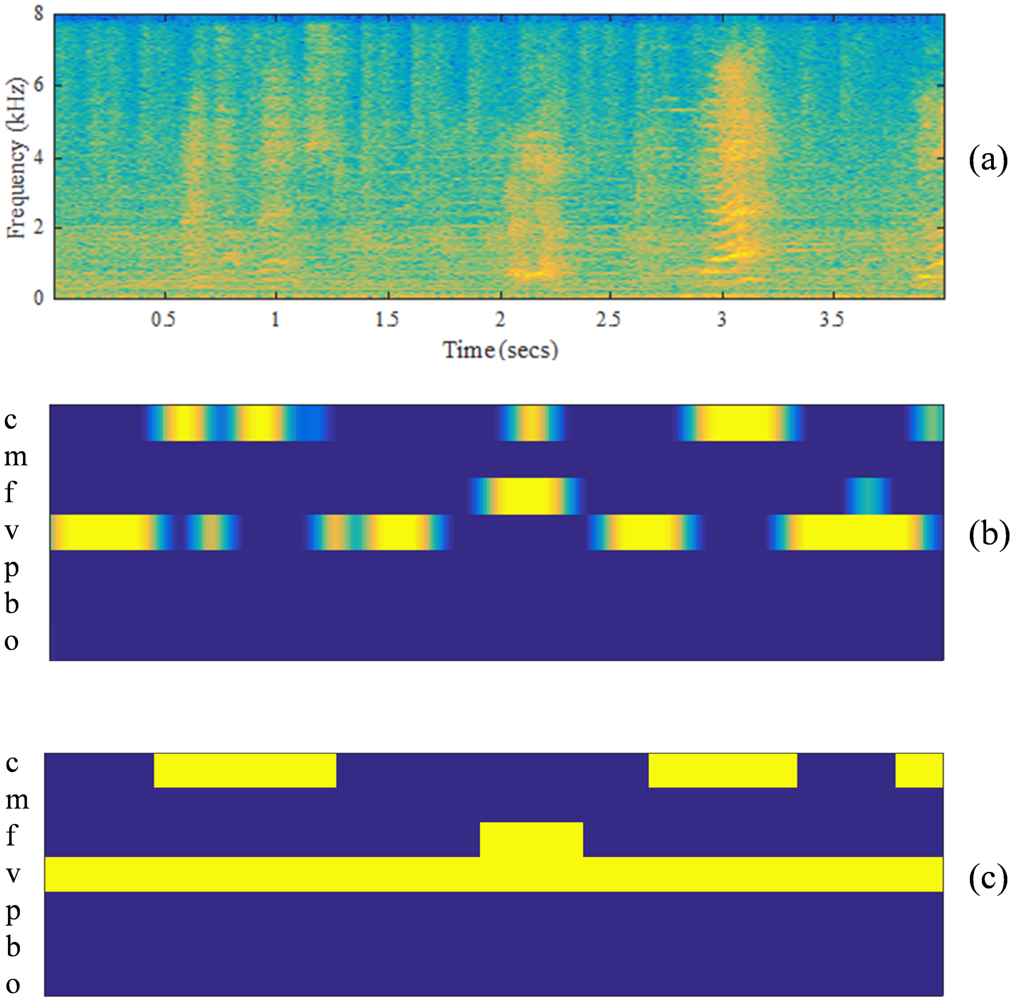

Figure 4.3: (a) Spectrogram of the audio clip. (b) Detector output of the LDGC model. (c) Ground truth label of the recording.

The LDGC model is also able to find timestamps of events in recordings. Fig. 4.3(a) shows the spectrogram which corresponds to a record in the evaluation dataset. The weak label for this recording includes children speech, female speech, and TV sounds. Fig. 4.3(b) shows the frame-level scores of the recording, which are activations from outputs of the convolutional layer after filtering. Fig. 4.3(c) shows the locations of the actual events in the recording. It is shown that the LDGC model detects the acoustic events successfully when the event is monophonic. However, it does not work well when the events are overlapped. This means that polyphonic events are not properly modeled in the proposed framework. Further research is required to make a robust model for polyphonic events.

## 4.4 Summary

We proposed a local detector and a global classifier framework for audio tagging. The local detector extracts distinct information, and the global classifier summarizes it to make a decision. Results from the experiments demonstrated that the proposed model outperformed the other models based on neural networks. The model is able to locate event positions without temporal annotations during training.

# Chapter 5

# Deep Convolutional Neural Network with Structured Prediction for Weakly Supervised Audio Event Detection

## 5.1 Introduction

People experience a variety of audio events with meaningful information that can be useful for human activities. AED aims to identify the occurrence of specific sounds in audio recordings. As the amount of multimedia data on the internet is growing rapidly, analyzing audio events will help describing and understanding environmental and social activities in video and audio contents. AED is also useful in many other applications, including surveillance, self-driving cars, healthcare, smart home systems, and military applications.

In early studies on AED, several approaches were proposed based on signal processing and machine learning techniques, and recently deep learning based methods have been widely developed. Most of these studies were based on fully supervised learning methods that require strongly labeled data. In strongly labeled data, either audio event examples are directly provided or the exact time of each audio event is given. However, building a large strongly labeled database is a time-consuming and challenging work. For these reasons, there exist only a few publicly available large-scale audio event datasets with strong labels.

Recently, there have been some studies on weakly supervised AED [23], [24], [33], [34]. These studies focus on learning AED models based on weakly labeled data that provides only the presence or absence of events in the recording. We can obtain weakly labeled datasets much easier than strongly labeled datasets. For example, we can collect videos uploaded on the internet and use the tags of the videos as weak labels. However, it is problematic to directly use this data for AED since the exact occurrence times of the events are not known, which makes it difficult to learn a model for segment-level predictions.

Most of the AED methods use spectro-temporal representations as input features. Since the spectro-temporal feature of an audio signal, such as log mel spectrogram, can be considered as a 2D image, computer vision techniques can be applied to AED. In recent works on computer vision, deep learning approaches including CNN models such as the residual network (ResNet) [35], the densely connected convolution network (DenseNet) [36], the squeeze-and-excitation network (SENet) [37] have shown impressive performance. Also, many studies report that better results are obtained by using structured prediction methods, which consider dependencies of each pixel-level output [38], [39].

Early works on AED focused on detecting audio events based on various machine learning techniques. Several approaches were proposed based on HMMs [6], [7]. In [7], GMM - HMM based modeling similar to speech recognition techniques was proposed to model audio events. SVM [8], [9], [10] and NMF [11], [12], [13] were also applied to AED in some studies. Bag of words representation was used to represent and detect audio events with various classifiers [40], [41]. In [18], the use of multi-label DNNs is proposed for detecting temporally overlapping audio events in realistic environments. Many works on AED have been proposed based on CNN [42], [43], [44], [45]. RNNs have been utilized for AED [46], and also in conjunction with DNNs or CNNs [47], [48]. However, increasing the size of a fully supervised deep learning model is difficult due to a lack of large-scale strongly labeled datasets. This limitation can be somewhat alleviated by model regularization and data augmentation, but it is difficult to overcome the limitation completely.

There have been several studies on analyzing and detecting audio events in a weakly supervised scenario. Weakly supervised AED has been widely studied after the release of AudioSet [4], which contains more than two million 10-second YouTube clips with weak audio labels. In the early studies of weakly supervised AED, a MIL [49] based approach was proposed in [23]. The authors formulated weakly supervised AED as a MIL problem and proposed MIL methods based on SVM and DNN. Although the training was done using weakly labeled data without temporal information, the authors showed that temporal localization of audio events was able to be extracted. In [24], the authors proposed a unified framework for SWSL using a graph-based model. The proposed model was able to be learned simultaneously from strongly and weakly labeled data.

Deep learning based methods have been widely proposed for weakly supervised AED and many of these methods have employed CNNs [33], [34], [50], [51]. In [33], CNN was applied with an event-specific Gaussian filter layer, which was designed to improve its learning ability. [34] proposed a CNN structure with adaptive pooling operators to aggregate temporally dynamic predictions. [50] used CNN to scan and produce outputs at small segments and then map these segment-level outputs to full recording level outputs. [51] used transfer learning to effectively convey knowledge from weakly labeled web audio data to the target data. In the DCASE 2017 [3], most of the top performing methods on the weakly labeled task relied on CNNs [52], [53], [54].

Recent improvements in computer hardware have enabled training very deep CNNs. However, this is not easy due to the problem of vanishing/exploding gradients particularly in lower layers. Many algorithms have been proposed to solve this problem such as ResNet [35]. ResNet introduces a residual block that sums a nonlinear transformation of the input and its identity mapping. The identity mapping is implemented through a shortcut connection which makes the networks avoid the vanishing gradient problem. The shortcut connections help to improve the performance of the networks and obtain faster convergence of training. As an extension of ResNets, a new CNN architecture, called DenseNet, was introduced in [36]. DenseNet is built from stacks of dense blocks and pooling operations. The dense blocks consist of multiple layers with direct connections from any layer to all subsequent layers to improve the information flow between layers.

In [37], the authors focused on the channel relationship and proposed a novel architectural unit, the squeeze-and-excitation (SE) block, that adaptively recalibrated channel-wise feature responses by explicitly modeling interdependencies between channels. They proposed to squeeze global spatial information into a channel descriptor and modeled channel-wise relationships using a lightweight gating mechanism. They demonstrated that SE blocks brought significant improvements in the performance of the state-of-the-art CNNs at a minimal additional computational cost.

CRFs have been employed to enforce structure consistency in semantic segmentation. In [55], a fully connected CRF was used to consider the structural properties of the segmentation outputs. More recently, deep learning models integrating the densely connected CRF were proposed in many studies. DeepLab [38] proposed deep CNNs with atrous convolution, which is convolution with upsampled filters, and combined the responses at the final layer with a fully connected CRF. In [39], an RNN was introduced to approximate the mean-field iterations of CRF optimization, allowing for end-to-end training of both the fully convolutional network and the RNN.

In this chapter, we propose a deep convolutional network based on DenseNet and SENet for weakly supervised AED. We take advantage of strengthening feature propagation from DenseNet and modeling channel-wise relationships by SENet. Also, the correlations among segments in recordings are considered through a recurrent neural network (RNN) and conditional random field (CRF) [1]. We evaluate our proposed method and compare its performance with a CNN based baseline approach. Empirical results show that the proposed method outperforms the baseline on the DCASE 2017 task 4 dataset.

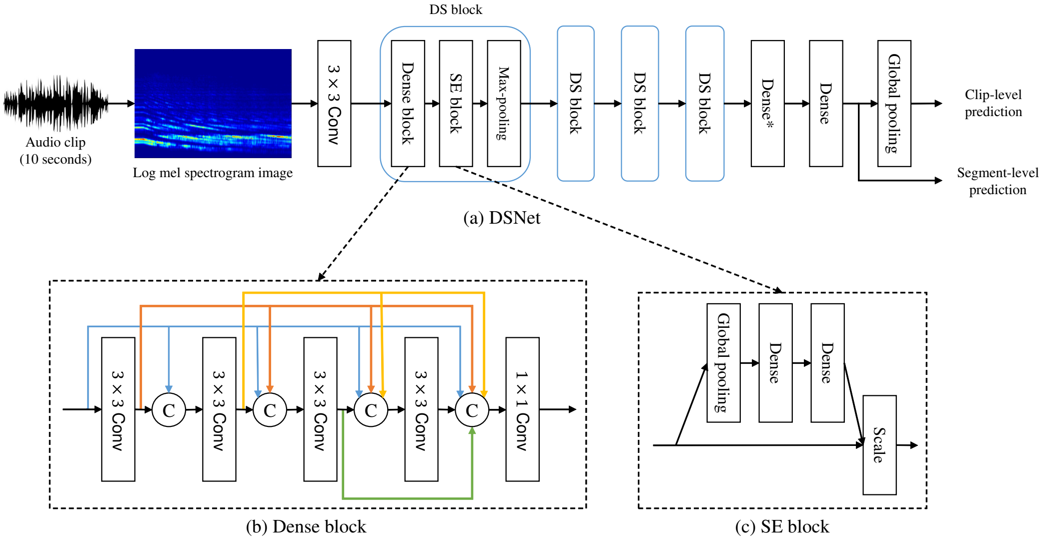

Figure 5.1: Overview of the proposed DSNet for weakly supervised AED. (**a**) The architecture of DSNet. The dense layer marked with * is replaced by a recurrent layer in DSNet-RNN. (**b**) The schema of the dense block. $C$ denotes a concatenation operation. (**c**) The schema of the SE block.

## 5.2 CNN with Structured Prediction for Weakly Supervised AED

In this section, we describe our weakly supervised AED model, referred to as DSNet. The overall structure of the DSNet is depicted in Fig. 5.1. The input of DSNet is a log mel spectrogram image $X \in R^{N \times M}$, where $N$ denotes the number of frames and $M$ is the number of mel filterbanks. First, convolution is performed on the log mel spectrogram images to extract feature maps. We use 4 DS blocks which consist of a dense block, an SE block, and a max-pooling layer. Two fully connected layers are applied for segment-level prediction. To detect overlapping audio events simultaneously, we have defined AED as a multi-label classification problem. For this, segment-level predictions are calculated using sigmoid activation functions at the final fully connected layer. A global pooling layer is applied for clip-level prediction. For structured prediction, DSNet with an RNN is proposed and a fully connected CRF is applied as a post-processing method. The detailed architectures of DSNet and DSNet-RNN are given in Section 5.3.

### 5.2.1 DenseNet

In a standard CNN, the output of the $l$th layer $\mathbf{o}_l$ is calculated by applying a non-linear transformation to the output of the previous layer $\mathbf{o}_{l-1}$

$$\mathbf{o}_l = L_l(\mathbf{o}_{l-1}) \tag{5.1}$$

where $L_l$ is a convolution followed by a non-linearity activation function. Conventional CNNs consist of a stack of convolutional layers. However, deeper CNNs are more difficult to train due to vanishing gradients. In ResNet [35], residual blocks are used to train deeply structured neural networks. A residual block sums the identity mapping of the input to the output of the layer. The output $\mathbf{o}_l$ of a residual block is given by

$$\mathbf{o}_l = H_l(\mathbf{o}_{l-1}) + \mathbf{o}_{l-1} \tag{5.2}$$

where $H_l$ is a non-linear transformation which usually consists of a single layer or a stack of multiple layers. The identity mapping acts like a skip connection from a lower layer to the upper layer, which enables input features to be reused and the gradient to flow directly from the upper layer to the lower layer.

DenseNet [36] is built from stacks of dense blocks. To improve the information flow between layers, DenseNet uses skip connections from any layer to all subsequent layers in each dense block. The output of each layer in dense blocks can be expressed as

$$\mathbf{o}_l = L_l([\mathbf{o}_{l-1}, ..., \mathbf{o}_0]) \tag{5.3}$$

where [ ] represents the concatenation of feature maps. DenseNet may look similar to ResNet, which introduces skip connections. However, this small modification makes a noticeable difference between the two networks. DenseNet is more efficient than ResNet in parameter usage. Thanks to short connections to all feature maps in the architecture, information from previously computed feature maps can be reused easily.

We use 4 convolutional layers and a single bottleneck layer in each dense block. To improve computational efficiency, the bottleneck layer compresses all feature maps in the dense block into a reduced number of feature maps using $1 \times 1$ convolution. For all convolutional layers in the model, each side of the inputs is zero-padded by one pixel to keep the feature map size fixed and batch normalization is applied before ReLU for better training performance. We use more feature maps on the upper dense blocks to compensate for feature map size reduction in each max-pooling layer.

### 5.2.2 Squeeze-and-Excitation

We use SE blocks [37] to consider interdependencies between channels. In the SE block, global spatial information is squeezed into a channel descriptor using global average pooling. A channel descriptor $z \in R^C$ is extracted by averaging the input feature map $U \in R^{H \times W \times C}$ through its spatial dimensions $H \times W$. To utilize the information aggregated in the squeeze operation, a simple gating mechanism is employed. The channel descriptor $z$ is transformed into a set of channel weights

$s \in R^C$ which is given by

$$s = \sigma(W_2 \delta(W_1 z)) \tag{5.4}$$

where $\sigma$ and $\delta$ respectively refer to sigmoid and rectified linear functions. To reduce model complexity and aid generalization, a bottleneck is formed by $W_1 \in R^{\frac{C}{r} \times C}$ and $W_2 \in R^{C \times \frac{C}{r}}$. We set the dimensionality-reduction ratio $r$ to 4 in our system. The final output of the SE block is obtained by scaling $U$ with channel weights $s$ for each channel. In this manner, channels possessing more important information can be emphasized.

### 5.2.3 Global Pooling for Aggregation

The proposed DSNet aims to predict both segment-level and clip-level labels. To train DSNet with only weak labels (clip-level labels), we need to aggregate segment-level predictions to form clip-level predictions. A common approach would be taking an average over all segment predictions corresponding to a clip-level prediction. In this approach, all segments of the clip have the same influence on the clip-level prediction. However, clips with a positive label can also contain negative segments which disturb the training process. In the multiple instance learning framework, a global max-pooling is applied to aggregate segment-level predictions into a clip-level prediction. In the max-pooling approach, the clip-level prediction focuses on the most positive segment in the clip and disturbance from negative segments can be reduced. However, with global max-pooling aggregation, only the most positive segment in each clip is active in training during backpropagation and other segments are ignored.

To take advantage of both methods, we apply the LogSumExp (LSE) function, which is a smooth approximation of the max function. The LSE function is given as

$$y_k = \frac{1}{\alpha} \log(\frac{1}{T} \sum_{i=1}^{T} \exp(\alpha * s_{i,k})) \tag{5.5}$$

where $y_k$ is a clip-level prediction for class $k$, $s_{t,k}$ is the segment label of the $i$th segment for class $k$ and $T$ is the number of segments in a clip. In (5.5), $\alpha$ is a hyperparameter to control the sharpness of the function. As $\alpha$ increases, the function approaches to the max function and as $\alpha$ decreases, the function approaches the average function. With the LSE pooling, we can use all the segments of the clip during training and also focus on positive segments in positive clips. We set the parameter $\alpha = 0.5$ in our system. To train our model with only weak labels, we apply the mean square error as the cost function, which is given by

$$C_{cl} = ||L - y||^2 \tag{5.6}$$

where $L$ denotes the true label, $y$ is the clip-level prediction.

### 5.2.4 Structured Prediction for Accurate Event Localization

#### RNN-based Structured Prediction

Segment-level prediction can be performed using DSNet described earlier. However, segment-level predictions may not be robust since DSNet does not make good use of long-term contextual information. Better segment-level prediction results can be obtained by considering long-term contextual information and incorporating prior knowledge into our model. To consider long-term dependency between segment predictions, an RNN is applied at the top of DSNet. We refer to DSNet with RNN as

DSNet-RNN. The structure of DSNet-RNN is almost the same with DSNet except that the dense layer marked with $*$ in Fig. 5.1 is replaced by a single layer RNN with bi-directional GRUs (Bi-GRUs) [56]. However, in weakly supervised learning, there is a lack of accurate label information on each segment. The incorrect information on each segment may affect other segments through the RNN. In order to mitigate this problem, it is desirable to train our model by applying some prior knowledge that audio events are generally continuous. To utilize this prior knowledge, we define a prediction smoothness cost $C_{ps}$ as

$$\begin{aligned} C_{ps} &= \sum_{i,j=1}^{T} \mu(s_i, s_j) u(i, j), \\ \mu(s_i, s_j) &= ||s_i - s_j||, \\ u(i, j) &= exp(-\frac{||p_i - p_j||^2}{2\sigma_{ps}^2}), \end{aligned} \tag{5.7}$$

where $s_i$ is the segment prediction of the $i$th segment and $p_i$ denotes its normalized temporal position. The prediction smoothness cost $C_{ps}$ encourages segment predictions to be continuous over time by penalizing nearby segments with different predictions. The cost function for training the DSNet-RNN is given by

$$C = C_{cl} + \lambda C_{ps} \tag{5.8}$$

where $\lambda$ is a compromising parameter.

### CRF Post-processing

As the proposed model can produce segment-level predictions, we can determine the border of audio events through post-processing. A common approach is to smooth the segment-level predictions and threshold them for boundary decisions. However, since this approach does not take dependency between the segments into account, it is not easy to determine the borders of audio events precisely. In order to address this issue, we apply CRF for post-processing the segment-level predictions. To reflect the full relationship among segments, we incorporate the fully connected CRF model proposed in [55] into our system.

In the conventional approach, segment-level predictions $s_i$ are smoothed and thresholded for segment-level classification. The threshold value $th_v$ is determined to have the best f-score on the validation set. In the CRF post-processing approach, label assignment probability of each class for the $i$th segment $P(i)$ is calculated as

$$P(i) = sigmoid((s_i - th_v)). \tag{5.9}$$

The energy function for the fully connected CRF is given as

$$E(x) = \sum_i \theta_i + \sum_{ij} \theta_{ij}, \tag{5.10}$$

$$\theta_i = -\log(P(i)), \tag{5.11}$$

$$\begin{aligned} \theta_{ij} = \mu(i,j) * [w_{mel} * \exp(-\frac{||m_i - m_j||^2}{2\sigma_{mel}^2}) \\ + w_{pos} * \exp(-\frac{||p_i - p_j||^2}{2\sigma_{pos}^2})], \end{aligned} \tag{5.12}$$

where $\theta_i$ represents the unary potential at the $i$th segment and $\theta_{ij}$ is the pairwise potential between the $i$th and $j$th segments. In the pairwise potential, $\mu(i,j) = 1$

if the $i$th and $j$th segments have different label assignments, and zero otherwise. $p_i$ denotes the temporal position and $m_i$ is the log mel spectrum of the $i$th segment. The hyperparameters $w_{mel}, \sigma_{mel}, w_{pos}, \sigma_{pos}$ control the Gaussian kernels. The pairwise potential penalizes segments with similar log mel spectra and positions having different labels. This model can efficiently infer the probabilities using mean field approximation and efficient message passing through high-dimensional filtering [55].

## 5.3 Experiments

### 5.3.1 Dataset

The DCASE 2017 task 4 Dataset [3] was published for the task of "Large-scale weakly supervised sound event detection for smart cars" in the DCASE 2017 challenge. The dataset employs a subset of AudioSet by Google [4]. The DCASE 2017 task 4 Dataset consists of 17 audio events divided into two categories: "Warning" and "Vehicle". The dataset contains audio classes for self-driving cars, smart cities and related areas. The dataset contains 51,172 clips of the training set, 488 clips of validation set, and 1,103 clips of evaluation set. Every clip is less than 10 seconds long. Each clip may correspond to more than one audio event and possibly has overlapping audio events. The dataset is obtained by collecting real-life recordings that contain noise and unknown class signals. The training set has weak labels denoting the presence of a given audio event in the clip, and no timestamps are provided. For the validation and evaluation sets, strong labels with timestamps are provided for the purpose of performance evaluation.

### 5.3.2 Feature Extraction

As inputs to the neural networks, we used log mel filterbank features. We extracted 128 mel bands from 0 Hz to 22050 Hz. We applied a window size of 1100 samples with a shift of 365 samples for frame segmentation to produce 800 frames in a 10-second clip. The logarithm of the mel band energies are calculated and each log mel energy was normalized by subtracting its mean and dividing by its standard deviation computed over the training set. As a result, an 800 × 128 normalized log mel spectrogram image was extracted for each 10-second clip.

### 5.3.3 DSNet and DSNet-RNN Structures

The specific configuration of the proposed model is described in Table 5.1. The extracted normalized log mel spectrogram image was used for input to the neural networks. A convolution layer was used to produce feature maps for dense blocks. These networks consisted of four dense blocks each with four convolution layers and one bottleneck layer. The convolution layers consisted of three consecutive operations: $3 \times 3$ convolution, batch normalization and ReLU. We used a $1 \times 1$ convolution layer to reduce channels. An SE block and a max-pooling layer were placed after each dense block. For segment-level prediction, two dense layers were applied in the DSNet, and Bi-GRU and a dense layer were applied in the DSNet-RNN. Finally, the segment-level predictions were aggregated through the global pooling layer for clip-level prediction. We set $\sigma_{ps} = 0.1$ in (5.7) and $\lambda = 0.01$ in (5.8) to train the DSNet-RNN. The parameter size of the DSNet is 0.32M, which is similar to that of the baseline CNN. The DSNet-RNN has more parameters than the others due to the Bi-GRUs used for structured prediction.

Table 5.1: DSNet and DSNet-RNN architectures

| Layers | output size | DSNet | DSNet-RNN |
|---|---|---|---|
| Convolution | 800×128×32 | [3×3, 32 conv] | |
| Dense block | 800×128×32 | [3×3, 16 conv]×4<br>[1×1, 32 conv] | |
| SE block | 800×128×32 | bottleneck size 8 | |
| Max-pooling | 800×64×32 | 1×2 max-pool | |
| Dense block | 800×64×48 | [3×3, 16 conv]×4<br>[1×1, 48 conv] | |
| SE block | 800×64×48 | bottleneck size 12 | |
| Max-pooling | 400×32×48 | 2×2 max-pool | |
| Dense block | 400×32×64 | [3×3, 16 conv]×4<br>[1×1, 64 conv] | |
| SE block | 400×32×64 | bottleneck size 16 | |
| Max-pooling | 200×16×64 | 2×2 max-pool | |
| Dense block | 200×16×64 | [3×3, 16 conv]×4<br>[1×1, 64 conv] | |
| SE block | 200×16×64 | bottleneck size 16 | |
| Max-pooling | 100×8×64 | 2×2 max-pool | |
| Reshape | 100×512 | 100×8×64 to 100×512 | |
| Segment-level prediction | 100×17 | 256 dense(ReLU)<br>17 dense(sigmoid) | 128 Bi-GRUs<br>17 dense(sigmoid) |
| Clip-level prediction | 17 | global LSE pooling | |
| Parameters | - | 0.32M | 0.69M |

Table 5.2: Baseline CNN architecture

| Layers | output size | CNN |
|---|---|---|
| Convolution | 800×128×32 | [3×3, 32 conv]×2 |
| Max-pooling | 800×64×32 | 1×2 max-pool |
| Convolution | 800×64×32 | [3×3, 32 conv]×2 |
| Max-pooling | 400×32×32 | 2×2 max-pool |
| Convolution | 400×32×64 | [3×3, 64 conv]×2 |
| Max-pooling | 200×16×64 | 2×2 max-pool |
| Convolution | 200×16×64 | [3×3, 64 conv]×2 |
| Max-pooling | 100×8×64 | 2×2 max-pool |
| Reshape | 100×512 | 100×8×64 to 100×512 |
| Segment-level prediction | 100×17 | 256 dense(ReLU)<br>17 dense(sigmoid) |
| Clip-level prediction | 17 | global LSE pooling |
| Parameters | - | 0.29M |

### 5.3.4 Baseline CNN Structure

To verify the performance of the proposed method, we compared the proposed method with a baseline model. In the DCASE 2017 challenge, several CNN-based models were proposed and showed good performance in weakly supervised AED [52], [53], [54]. We chose a CNN baseline model similar to the models proposed in the DCASE 2017 Challenge. The specific configuration of the baseline model is described in Table 5.2. The audio feature for the baseline was the same as that of the proposed model, a $800 \times 128$ normalized log mel spectrogram image. The baseline model consisted of four stacks of two convolution layers and a max-pooling layer. The last max-pooling layer was connected to two dense layers to produce segment-level predictions, and the segment-level predictions were aggregated in the global pooling layer.

### 5.3.5 Training and Evaluation

The neural network models were implemented using TensorFlow [31]. We set the hyperparameters such that they provided the highest segmental f-score on the validation set. All networks were trained with Adam [32]. A dropout [21] rate at 0.1 is applied to the output of the SE blocks and the dense layer with ReLU. We used mini-batches of 10 clips and a learning rate of 0.0001. We used the validation set to earlystop the training based on the segmental f-score. To deal with the unbalance between classes on the training set, we applied undersampling to the classes with more than 1000 clips. The networks were trained on NVIDIA Tesla M40 GPUs.

For evaluation, the optimal thresholds were selected to have the best performance on the validation set. The segment-level predictions were smoothed with a Hanning window of length 41 before thresholding. We set the CRF parameters in (5.12) to $w_{mel} = 1$, $\sigma_{mel} = 1$, $w_{pos} = 1$ and $\sigma_{pos} = 25$, which showed the best segment-level f-score on the validation set. To perform multi-labeled classification, CRF post-processing was performed separately for each class. We employed 10 mean field iterations in the test phase.

### 5.3.6 Metrics

In our work, both clip-level and segment-level evaluation metrics were used. The default segment length used in this work was 100 ms, which is shorter than the segment length used in the DCASE challenge, 1 second. This was because our system aims to detect audio events accurately in time via structured prediction. Since the dataset for evaluation has multi-label annotations, we used f-score. For segment-level evaluation, segment-based ER was also measured. A detailed explanation of both

Table 5.3: Clip-level results on the DCASE 2017 task 4 evaluation set

| Model | F | P | R |
|---|---|---|---|
| CNN | 0.5506 | 0.5667 | 0.5353 |
| DSNet | 0.5853 | 0.5822 | 0.5883 |
| DSNet-RNN | 0.5839 | 0.5504 | 0.6281 |

Table 5.4: Class-wise clip-level f-score results

| Class | CNN | DSNet | DSNet-RNN |
|---|---|---|---|
| Train horn | 0.5273 | 0.4615 | 0.5102 |
| Air horn, truck horn | 0.4000 | 0.5455 | 0.5783 |
| Car alarm | 0.4267 | 0.4500 | 0.3836 |
| Reversing beeps | 0.3373 | 0.3765 | 0.4186 |
| Ambulance | 0.5556 | 0.4681 | 0.4854 |
| Police car | 0.4906 | 0.5778 | 0.6525 |
| Fire engine, fire truck | 0.5606 | 0.6055 | 0.5586 |
| Civil defense siren | 0.7704 | 0.8160 | 0.8189 |
| Screaming | 0.6833 | 0.7059 | 0.8333 |
| Bicycle | 0.4675 | 0.4615 | 0.3294 |
| Skateboard | 0.5946 | 0.7627 | 0.6372 |
| Car | 0.6266 | 0.6759 | 0.6411 |
| Car passing by | 0.2727 | 0.2931 | 0.2468 |
| Bus | 0.4238 | 0.4000 | 0.2637 |
| Truck | 0.4455 | 0.4541 | 0.4505 |
| Motorcycle | 0.5465 | 0.6324 | 0.7009 |
| Train | 0.7209 | 0.7883 | 0.7759 |

evaluation metrics is described in Chapter 2.

### 5.3.7 Results and Discussion

#### Audio Tagging

Table 5.3 presents the clip-level tagging results on the DCASE 2017 task 4 evaluation set and parameter sizes of each model. The results show that the DSNet has an absolute improvement of 0.0347 over the baseline CNN in terms of f-score. The

Table 5.5: Segment-level results on the DCASE 2017 task 4 evaluation set

| Model | F | P | R | ER |
|---|---|---|---|---|
| CNN | 0.4987 | 0.4598 | 0.5447 | 0.7568 |
| DSNet | 0.5135 | 0.4746 | 0.5593 | 0.7039 |
| DSNet-RNN | 0.5354 | 0.5074 | 0.5667 | 0.6213 |

performance of the DSNet indicates that DenseNet and SENet are suitable not only for image processing but also for audio processing. The DSNet-RNN shows almost the same performance as the DSNet in clip-level metrics, which means the structured prediction has little effect on the clip-level performance.

The class-wise f-score results for the CNN, DSNet and DSNet-RNN models are presented in Table 5.4. While there is some variation across classes, the DSNet and DSNet-RNN show better performance than CNN on most classes. The performance of the DSNet is considerably better compared to the baseline CNN for the "air horn, truck horn", "police car", "skateboard" and "motorcycle" classes and the DSNet-RNN shows better performance than the baseline CNN in the "air horn, truck horn", "police car", "screaming" and "motorcycle" classes. The best performing class for all models is "civil defense siren" which consists of long and high volume sounds and the worst performing class is "car passing by", which consists of short and low volume sounds.

### Event Detection with Localization

Table 5.5 presents the segment-level results on the DCASE 2017 task 4 evaluation set. Both the DSNet and DSNet-RNN outperform the baseline CNN model in f-score by 0.0148 and 0.0367, respectively. Similar to the clip-level results, the DSNet performs better than conventional CNN by using DenseNet and SENet. Especially,

Table 5.6: Segment-level results for the DSNet-RNN at different $\lambda$

| $\lambda$ | F | ER |
|---|---|---|
| 0 | 0.5168 | 0.6564 |
| 0.005 | 0.5184 | 0.7048 |
| 0.01 | 0.5354 | 0.6213 |
| 0.02 | 0.5281 | 0.6867 |
| 0.05 | 0.5039 | 0.8109 |

Table 5.7: Effect of CRF on segment-level performance

| | Before CRF | | After CRF | |
|---|---|---|---|---|
| Model | F | ER | F | ER |
| CNN | 0.4987 | 0.7568 | 0.5195 | 0.6680 |
| DSNet | 0.5135 | 0.7039 | 0.5265 | 0.6849 |
| DSNet-RNN | 0.5354 | 0.6213 | 0.5432 | 0.6131 |

the DSNet-RNN shows the best performance in segment-level results. This indicates that each segment-level prediction benefits from considering contextual information in the neural network.

The weight $\lambda$ introduced in (5.8) is a hyperparameter which allows us to control the dependency of the cost function on structured prediction. The effect of the weight $\lambda$ on the DSNet-RNN is presented in Table 5.6. The result shows that when $\lambda = 0$, the model does not show significant performance improvement over the DSNet. This means that the flow of uncertain information in the RNN may hinder the training of the model in weakly supervised learning. Overall results show that the performance of the model can be improved by restricting uncertain information flow with appropriate constraints based on prior information. The model showed the best performance when $\lambda = 0.01$, which is a default value used when training the DSNet-RNN.

Table 5.7 presents the performance of CRF post-processing on the segment-level

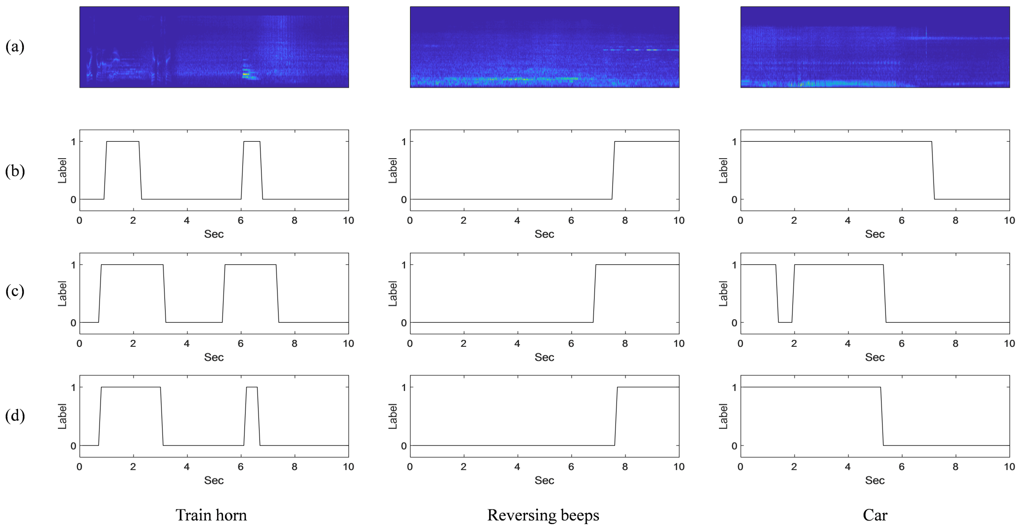

Figure 5.2: The results of the DSNet-RNN before and after CRF. (**a**) Log mel spectrograms of audio events. (**b**) Segment-level ground truth labels. (**c**) Predicted segment-level labels before CRF. (**d**) Predicted segment-level labels after CRF.

performances. All the models show performance improvement through CRF post-processing. In the DSNet-RNN, the performance improvement is relatively low. This indicates that the DSNet-RNN already reflected contextual information and hence the additional benefit from CRF post-processing is relatively small. The results of the DSNet-RNN with and without CRF post-processing are visualized in Fig. 5.2. By employing CRF post-processing, we can correct isolated inaccurate predictions and improve the accuracy of the predictions particularly in the boundaries of the events.

### 5.3.8 Comparison with the DCASE 2017 task 4 Results

For comparison, the results of our models and the top results from the DCASE 2017 task 4 are presented in Table 5.8. In the DCASE 2017 task 4, Xu et al. [52]

Table 5.8: Comparison with the DCASE 2017 results on evaluation set

| Model | $F_{tag}$ | $F_{1s}$ | $ER_{1s}$ |
|---|---|---|---|
| DSNet | **0.585** | 0.530 | 0.689 |
| DSNet+CRF | **0.585** | 0.542 | 0.644 |
| DSNet-RNN | **0.584** | **0.550** | 0.606 |
| DSNet-RNN+CRF | **0.584** | **0.557** | **0.570** |
| Xu *et al.* [52] | 0.556 | 0.518 | 0.730 |
| Lee *et al.* [53] | 0.526 | **0.555** | 0.660 |

and Lee et al. [53] showed the best performance in audio tagging (clip-level) and sound event detection (segment-level), respectively. Xu et al. [52] used the learnable gated activation function in their model and Lee et al. [53] used CNNs with multiple scale input. Both of them also used the fusion or ensemble of models for the better detection performance.

For a fair comparison, we compared the segment-level results of our proposed model in 1 second time resolution. Our models showed better performance in both clip-level and segment-level results, even without the fusion or ensemble of models. The proposed models outperformed Xu et al. [52] in clip-level f-score. In the segment-level metrics, the DSNet-RNN achieved a similar performance as Lee et al. [53] in f-score and showed a better performance in ER.

## 5.4 Summary

In this chapter, we proposed DSNet, which is a combination of DenseNet and SENet, for weakly supervised AED. DSNet allows better information and gradient flow through direct connections between any two layers in dense blocks and adaptively recalibrates channel-wise feature responses using SE blocks. Moreover, we proposed a structured prediction framework and adopted it to DSNet. DSNet-

RNN utilizes contextual information while minimizing the propagation of uncertainty and CRF post-processing helps to refine segment-level predictions. Experiments showed that DSNet with structured prediction achieved state-of-the-art results in the DCASE 2017 task 4 dataset.

# Chapter 6

# Conclusions

In this thesis, some of the approaches for data-efficient and weakly supervised systems are proposed. Conventional AED models are trained using approaches based on supervised learning. For supervised learning, strongly labeled data is required. However, collecting large-scale strongly labeled data of audio events is challenging due to the diversity of audio event types and labeling difficulties. To overcome this problem, a data-efficient AED approach and weakly supervised approaches are proposed. The proposed approaches are based on deep learning methods and can be applied to different theoretical cases for audio event detection.

Firstly, we have proposed a data-efficient DNN-based AED system. In the proposed system, data augmentation is performed to deal with the data sparsity problem in the small training dataset and generate polyphonic event examples. An exemplar-based noise reduction algorithm is proposed for feature enhancement. For polyphonic event detection, a multi-labeled DNN classifier is employed. An adaptive thresholding algorithm is applied as post-processing for robust event detection in noisy conditions. From the experimental results, the proposed algorithm has shown promising performance for AED on a low-resource dataset.

Secondly, we have proposed an audio tagging system on weakly supervised data, which is labeled with only the existence of events. The proposed model is based on CNNs and consists of a local detector and a global classifier. The local detector detects local audio words that contain distinct characteristics of events, and the global classifier summarizes the information to make a decision on the recording. From the experimental results, we have found that the proposed model outperforms conventional artificial neural networks.

Finally, we have proposed an AED model based on DenseNet and SENet for weakly supervised AED. The proposed model allows better information and gradient flow through direct connections between any two layers in dense blocks and adaptively recalibrates channel-wise feature responses using SE blocks. We take advantage of strengthening feature propagation from DenseNet and modeling channel-wise relationships by SENet. Also, the correlations among segments in audio recordings are represented by RNN and CRF. Contextual information is utilized by RNN, and CRF post-processing helps to refine segment-level predictions. We evaluate our proposed method and compare its performance with a CNN based baseline approach. From a number of experiments, it has been shown that the proposed method is effective both on audio tagging and weakly supervised AED.

# Bibliography

[1] J. Lafferty, A. McCallum, and F. C. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *Proceedings of the 18th International Conference on Machine Learning (ICML)*, 2001, pp. 282–289.

[2] 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 *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2015, pp. 1–5.

[3] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, “Dcase 2017 challenge setup: Tasks, datasets and baseline system,” in *Proceedings of DCASE2017 Workshop*, 2017.

[4] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[5] A. Mesaros, T. Heittola, and T. Virtanen, “Metrics for polyphonic sound event detection,” *Applied Sciences*, vol. 6, no. 6, pp. 1–17, 2016.

[6] X. Zhou, X. Zhuang, M. Liu, H. Tang, M. Hasegawa-Johnson, and T. Huang, “Hmm-based acoustic event detection with adaboost feature selection,” *Multimodal technologies for perception of humans*, pp. 345–353, 2008.

[7] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, “Acoustic event detection in real life recordings,” in *2010 18th European Signal Processing Conference (EUSIPCO)*, Aug 2010, pp. 1267–1271.

[8] A. Temko and C. Nadeu, “Classification of acoustic events using svm-based clustering schemes,” *Pattern Recognition*, vol. 39, no. 4, pp. 682–694, 2006.

[9] A. Temko and C. Nadeu, “Acoustic event detection in meeting-room environments,” *Pattern Recognition Letters*, vol. 30, no. 14, pp. 1281–1288, 2009.

[10] J. Portelo, M. Bugalho, I. Trancoso, J. Neto, A. Abad, and A. Serralheiro, “Non-speech audio event detection,” in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2009, pp. 1973–1976.

[11] M. L. Chin and J. J. Burred, “Audio event detection based on layered symbolic sequence representations,” in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 1953–1956.

[12] J. F. Gemmeke, L. Vuegen, P. Karsmakers, B. Vanrumste, and H. V. hamme, “An exemplar-based nmf approach to audio event detection,” in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct 2013, pp. 1–4.

[13] A. Mesaros, T. Heittola, O. Dikmen, and T. Virtanen, "Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 151–155.

[14] 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.

[15] J. Dennis, H. D. Tran, and E. S. Chng, "Overlapping sound event recognition using local spectrogram features and the generalised hough transform," *Pattern Recognition Letters*, vol. 34, no. 9, pp. 1085–1093, 2013.

[16] T. Heittola, A. Mesaros, T. Virtanen, and M. Gabbouj, "Supervised model training for overlapping sound events based on unsupervised source separation." in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 8677–8681.

[17] Z. Kons, O. Toledo-Ronen, and M. Carmel, "Audio event classification using deep neural networks." in *Interspeech*, 2013, pp. 1482–1486.

[18] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1–7.

[19] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 9, pp. 1469–1477, 2015.

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

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[22] Q. Kong, I. Sobieraj, W. Wang, and M. Plumbley, “Deep neural network baseline for dcase challenge 2016,” *Proceedings of DCASE 2016*, 2016.

[23] A. Kumar and B. Raj, “Audio event detection using weakly labeled data,” in *Proceedings of the 2016 ACM on Multimedia Conference*, 2016, pp. 1038–1047.

[24] A. Kumar and B. Raj, “Audio event and scene recognition: A unified approach using strongly and weakly labeled data,” *arXiv preprint arXiv:1611.04871*, 2016.

[25] A. Mesaros, T. Heittola, and T. Virtanen, “Tut database for acoustic scene classification and sound event detection,” in *2016 24th European Signal Processing Conference (EUSIPCO)*, 2016, pp. 1128–1132.

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

[27] Q. Kong, I. Sobieraj, W. Wang, and M. D. Plumbley, “Deep neural network baseline for DCASE challenge 2016,” DCASE2016 Challenge, Tech. Rep., September 2016.

[28] T. H. Vu and J.-C. Wang, "Acoustic scene and event recognition using recurrent neural networks," DCASE2016 Challenge, Tech. Rep., September 2016.

[29] T.-W. Su, J.-Y. Liu, and Y.-H. Yang, "Weakly-supervised audio event detection using event-specific gaussian filters and fully convolutional networks," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[30] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "A joint detection-classification model for audio tagging of weakly labelled data," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[31] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

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

[33] T.-W. Su, J.-Y. Liu, and Y.-H. Yang, "Weakly-supervised audio event detection using event-specific gaussian filters and fully convolutional networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 791–795.

[34] B. McFee, J. Salamon, and J. P. Bello, "Adaptive pooling operators for weakly labeled sound event detection," *arXiv preprint arXiv:1804.10070*, 2018.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

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

[37] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[38] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[39] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, “Conditional random fields as recurrent neural networks,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1529–1537.

[40] S. Pancoast and M. Akbacak, “Bag-of-audio-words approach for multimedia event classification,” in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012, pp. 2105–2108.

[41] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, “Sparse representation based on a bag of spectral exemplars for acoustic event detection,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 6255–6259.

[42] H. Zhang, I. McLoughlin, and Y. Song, “Robust sound event recognition using convolutional neural networks,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 559–563.

[43] H. Phan, L. Hertel, M. Maass, and A. Mertins, “Robust audio event recognition with 1-max pooling convolutional neural networks,” *arXiv preprint arXiv:1604.06338*, 2016.

[44] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, T. Virtanen, E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, “Convolutional recurrent neural networks for polyphonic sound event detection,” *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 25, no. 6, pp. 1291–1303, 2017.

[45] 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.

[46] T. Hayashi, S. Watanabe, T. Toda, T. Hori, J. Le Roux, and 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, 2017.

[47] J. Lee, T. Kim, J. Park, and J. Nam, “Raw waveform-based audio classification using sample-level cnn architectures,” *arXiv preprint arXiv:1712.00866*, 2017.

[48] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, “Sound event localization and detection of overlapping sources using convolutional recurrent neural

networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 34–48, 2018.

[49] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning," in *Advances in neural information processing systems*, 1998, pp. 570–576.

[50] A. Kumar and B. Raj, "Deep cnn framework for audio event recognition using weakly labeled web data," *arXiv preprint arXiv:1707.02530*, 2017.

[51] A. Kumar, M. Khadkevich, and C. Fügen, "Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 326–330.

[52] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Surrey-CVSSP system for DCASE2017 challenge task4," DCASE2017 Challenge, Tech. Rep., September 2017.

[53] D. Lee, S. Lee, Y. Han, and K. Lee, "Ensemble of convolutional neural networks for weakly-supervised sound event detection using multiple scale input," DCASE2017 Challenge, Tech. Rep., September 2017.

[54] J. Lee, J. Park, and J. Nam, "Combining multi-scale features using sample-level deep convolutional neural networks for weakly supervised sound event detection," DCASE2017 Challenge, Tech. Rep., September 2017.

[55] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Advances in neural information processing systems*, 2011, pp. 109–117.

[56] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.

# 요 약

일반적인 음향 이벤트 탐지 시스템은 교사학습을 통해 훈련된다. 교사학습을 위해서는 강한 레이블 데이터가 요구된다. 하지만 강한 레이블 데이터는 음향 이벤트의 다양성 및 레이블의 난이도로 인해 큰 데이터베이스를 구축하기 어렵다는 문제가 있다. 본 논문에서는 이러한 문제를 해결하기 위해 음향 이벤트 탐지를 위한 데이터 효율적 활용 및 약한 교사학습 기법에 대해 제안한다.

첫 번째 접근법으로서, 데이터 효율적인 음향 이벤트 탐지 시스템을 제안한다. 제안된 시스템에서는 데이터 증대 기법을 사용해 데이터 희소성 문제에 대응하고 중첩 이벤트 데이터를 생성하였다. 특징 벡터 향상을 위해 잡음 억제 기법이 사용되었고 중첩 음향 이벤트 탐지를 위해 다중 레이블 심층 인공신경망(DNN) 분류기가 사용되었다. 실험 결과, 제안된 알고리즘은 불충분한 데이터에서도 우수한 음향 이벤트 탐지 성능을 나타내었다.

두 번째 접근법으로서, 컨볼루션 신경망(CNN) 기반 오디오 태깅 시스템을 제안한다. 제안된 모델은 로컬 검출기와 글로벌 분류기로 구성된다. 로컬 검출기는 고유한 음향 이벤트 특성을 포함하는 로컬 오디오 단어를 감지하고 글로벌 분류기는 탐지된 정보를 요약하여 오디오 이벤트를 예측한다. 실험 결과, 제안된 모델이 기존 인공신경망 기법보다 우수한 성능을 나타내었다.

마지막 접근법으로서, 약한 교사학습 음향 이벤트 탐지 모델을 제안한다. 제안된 모델은 DenseNet의 구조를 활용하여 정보의 원활한 흐름을 가능하게 하고 SENet을

활용해 채널간의 상관관계를 모델링 한다. 또한, 오디오 신호에서 부분 간의 상관관계 정보를 재순환 신경망(RNN) 및 조건부 무작위 필드(CRF)를 사용해 활용하였다. 여러 실험을 통해 제안된 모델이 기존 CNN 기반 기법보다 오디오 태깅 및 음향 이벤트 탐지 모두에서 더 나은 성능을 나타냄을 보였다.

**주요어:** 음향 이벤트 탐지, 데이터 효율적 기법, 약한 교사학습, 딥 러닝

**학 번:** 2012-23248

# 감사의 글

대학원 입학이 엊그제 같은데, 어느덧 시간이 지나 졸업을 준비하게 되었습니다. 짧게도 길게도 느껴진 대학원 생활 동안 많은 분의 도움을 받아 뜻깊은 시간을 보낼 수 있었습니다. 그 시간을 돌이켜 보며 저에게 도움을 주신 많은 분께 간략하게나마 감사 인사를 드리려 합니다.

먼저 지도교수님이신 김남수 교수님께 감사의 말씀을 전합니다. 연구하면서 여러 어려운 순간들이 많았지만, 교수님의 지도와 조언으로 연구의 방향을 잡고 어려움을 이겨낼 수 있었습니다. 교수님은 연구 내적으로도 외적으로도 항상 모범이 되는 모습으로 저에게 많은 가르침을 주셨습니다. 관심과 배려로 저와 연구실 동료들 모두를 이끌어 주신 교수님께 다시 한번 감사의 말씀을 드립니다. 그리고 많은 조언과 지도로 제 박사 학위 논문에 많은 도움을 주신 김성철 교수님, 심병효 교수님, 장준혁 교수님, 신종원 교수님께 감사의 말씀을 드립니다.

연구실에서 오랜 시간 함께한 휴먼인터페이스 연구실 동료들에게도 감사의 인사를 전합니다. 짧은 시간이었지만 저에게 잘 대해 주시고 도움을 주신 창우형과 준식이형 감사드립니다. 기호형과 유광이형은 연구실 생활이 아직 낯설던 저를 많이 도와주셨습니다. 비록 방이 달라 많은 이야기를 나누지는 못했지만 두화형, 신재형, 현우형도 선배로서 저에게 많은 도움을 주셨습니다. 제 옆자리에서 많은 조언과 도움을 주셨던 철민이형과 기수형 덕분에 많은 것을 배울 수 있었습니다. 저와 다른 연구원들에게 많은 조언을 해 준 태균이, 항상 밝고 남을 잘 도와주던 석재, 운동도 연구도 열심히

한 강현이에게도 많은 도움을 받았습니다. 길호형도 회사 일로 바쁘셔서 자주 뵙지는 못했지만 좋은 성과 이루셨으면 좋겠습니다. 졸업 동기이자 연구에서도 가까웠던 수현이형은 서로에게 가장 도움을 많이 주고받았던 것 같습니다. 함께 사회로 나가게 되는데 앞으로도 서로 도움 주며 잘 지냈으면 좋겠습니다. 나보다 어리지만 배울 점이 많았던 동기 Sukanya는 어디서든 잘 해나갈 거라 믿습니다.

방장으로서 연구실에 많은 이바지를 한 준엽이, 연구실 일을 자기 일처럼 열심히 도와준 정훈이는 정말 고마운 후배들입니다. 연구실을 위해 노력한 만큼 둘 다 자신의 연구에서 좋은 성과 내리라 믿습니다. 차분하게 다른 사람의 이야기를 잘 들어주는 성준이와 주변 사람들을 잘 챙기는 연구실의 인기인 형용이에게는 여러 모로 많은 도움을 받았습니다. 둘 모두 머지않아 좋은 연구 결실을 보리라 생각합니다. 항상 진지하게 연구에 임하는 우현이, 새로운 길을 개척하고 있는 원익이, 연구실의 새 리더이자 인식의 수장인 현승이는 지금까지도 잘 해왔고 앞으로도 잘할 거라 믿습니다. 후배들이지만 먼저 사회에 나가 활약하는 세영이, 지환이, 석완이는 사회라는 정글에서 힘든 일도 있겠지만 잘 극복하고 날개를 펼치리라 믿습니다. 많은 경험으로 저에게 조언을 주신 주현이형, 회사 생활과 학업 병행이 바쁘고 힘드시겠지만 시간 잘 활용하셔서 좋은 성과 내시기 바랍니다. 행복한 신혼 보내는 새신랑 병진이, 묵묵하지만 꾸준한 성환이, 침착하고 성실한 민현이는 연구실의 든든한 기둥이 될 거라 생각합니다. 연구실 대표 춤꾼 형래는 졸업 준비 잘해서 무사히 졸업하길 바라고 연구실 홍일점으로서 알게 모르게 고생이 많을 지원이, 원익이와 함께 연구실의 NLP를 개척해 나갈 석민이는 씩씩하게 연구실 생활 잘 해나갈 거라 생각합니다. 많은 시간을 함께하진 못했지만, 연구실 뉴페이스 민찬이, 형주, 범준이, 병찬이는 자신에 맞는 연구주제를 잘 선택해 열심히 대학원 생활 보냈으면 좋겠습니다.

마지막으로 항상 저에게 힘이 되어준 가족들에게 감사를 전합니다. 언제나 저에게 넘치는 사랑과 믿음을 주신 아버지와 어머니, 그리고 항상 나를 물심양면으로 챙겨준 누나에게 다시 한번 진심으로 감사드립니다.