Ph.D. DISSERTATION

Learning-based sound source classification and localization with limited data

제한된 데이터를 통한 학습 기반 음원 종류 및 위치 추정

BY

Seungjun Lee

AUGUST 2022

DEPARTMENT OF NAVAL ARCHITECTURE AND
OCEAN ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

Learning-based sound source classification and localization with limited data

제한된 데이터를 통한 학습 기반 음원 종류 및 위치 추정

BY

Seungjun Lee

AUGUST 2022

DEPARTMENT OF NAVAL ARCHITECTURE AND
OCEAN ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Learning-based sound source classification and localization with limited data

제한된 데이터를 통한 학습 기반 음원 종류 및 위치 추정

지도교수 성 우 제

이 논문을 공학박사 학위논문으로 제출함

2022년 8월

서울대학교 대학원

조선해양공학과

이 승 준

이승준의 공학박사 학위 논문을 인준함

2022년 8월

위 원 장	(인)
부위원장	(인)
위 원	(인)
위 원	(인)
위 원	(인)

Abstract

Identifying the type and position of sound is one of the most important issues in the field of acoustics. In particular, we have no choice but to rely on acoustic information since visual information is strictly blocked within a real-world building structure to identify sources that can cause critical problems, such as mechanical defections. However, traditional approaches for sound source classification and localization utilize classic array processing techniques which are not applicable to sounds from real-world complex structures where sounds do not strictly follow the theory without carefully designed experiments. Therefore, we propose a learning-based approach to identify the type and position of sounds using a single microphone in a real-world building. We attempt to treat this problem as a joint classification problem in which we predict the exact positions of sounds while classifying the types that are assumed to be from predefined types of sounds. The most problematic issue is that while the types are readily classified under supervised learning frameworks with one-hot encoded labels, it is difficult to predict the exact positions of the sound from unseen positions during training. In order to address this potential discrepancy, we formulate the position identification problem as a zero-shot learning problem inspired by the human ability to perceive new concepts from previously learned concepts. We extract feature representations from audio data and vectorize the type and position of the sound source as 'type/position-aware attributes', instead of labeling each class with a simple one-hot vector. We then train a promising generative model to bridge the extracted features and the attributes by learning the class-invariant mapping to transfer the knowledge from seen to unseen classes through their attributes; generative adversarial networks are conditioned on the class-embeddings. Our proposed methods are evaluated on an indoor noise dataset, SNU-B36-EX, a real-world dataset collected inside a building.

keywords: sound classification, sound source localization, zero-shot learning, generative adversarial networks

student number: 2017-28959

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Unidentified sounds caused by people in a building are annoying to other residents and what is more, caused by some machinery or anomalies in an industrial factory should be identified and resolved where the sounds sometimes indicate critical mechanical defects. These sounds can be propagated along with the building structure, including walls, floors, ceilings, and columns, resulting in the propagation of this noise to all residents. Since there are complex structures with many rooms and floors in the real building structure, the visual information of the source might be blocked from their structure, while only annoying sounds are resonant. This often makes it difficult for residents to identify the source of the annoying sounds in such a real building. Therefore, an efficient system to identify the types and positions of the source using auditory information is necessary to mitigate the annoying problems.

There are two main research areas related to this problem: sound classification [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18] and sound source localization (SSL) [19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36]. As our problem is the combined form of sound classification and sound source localization, it can be loosely stated as the sound event localization and detection (SELD) problem, which has emerged in recent years [37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48]. Most existing studies on localizing a sound source from SSL or even SELD

have been conducted using multiple sensor networks and microphone arrays. However, such equipment is expensive to install with carefully designed settings, and it is not usual for a common person to have such equipment in a real-life building. In addition, the experimental settings of that literature only considered the sounds within the same environment [21, 22, 23, 26, 27, 28], whereas in a real building the sounds also arise from other rooms or floors. This means that we cannot easily take advantage of reverberation or multipath effects.

In recent years when collected data has been growing in size, neural networks have recently begun to draw attention for working well on not only matching complicated patterns from training data but also validation data from the similar data distribution [49]. The expressiveness of neural networks with a huge amount of parameters has overwhelmed other traditional methods when a huge amount of data and sufficient computing power are available, which leads to the great success of modern deep learning in many fields, such as image processing [50, 51], signal processing [52, 53], and natural language processing [54, 55]. For a practical approach to our problem instead of the traditional approaches, we consider a learning-based model for classifying the sounds based on their types and positions using a single microphone that is built into a mobile device or portable audio recorder.

In [56], the annoying sounds were successfully classified based on their types and positions using a supervised learning framework. However, in practice, the sound source can potentially be located in a continuous space, which might require the number of classes for training uncountable. Furthermore, significant effort is required to collect data to add new categories in real circumstances. Also, there are places within a building, such as restricted areas, where the data collection is often limited in real circumstances. Thus, the main focus of this study is to understand if the positions of the sound sources can be robustly predicted even for the points that were not seen during training.

These facts can be a notorious bottleneck to constructing an appropriate model

with a conventional supervised learning approach. Therefore, efficient learning frameworks are required to extract the transferable knowledge from previously available data to make a model generalize well on new data from new positions which are even very limited. One way to mitigate these problems is to use the zero-shot learning (ZSL) framework [57, 58], which is inspired by the human ability to perceive new semantics or concepts from what one has seen and learned previously. In general, ZSL models focus on learning the mapping function between the feature representations of data and the corresponding class-embeddings [59, 60, 61, 63, 64, 62, 65]. Through the learning process, the models are expected to capture the class-invariant mapping function between them. Therefore, the learned ZSL models can classify new examples that were unseen during training.

Therefore, for learning-based sound source classification and localization, we formulate the learning shared representation over the system inputs as a zero-shot learning problem, training the model to be generalized well on novel data from a new sound source. As the source localization problem can be considered as predicting the position of new data from the new sound source even unseen during the training, we apply the zero-shot learning framework to validate the methods on the real-world datasets, SNU-B36-EX, for source localization and classification problems. In summary, our problem is one of resolving sound classification and localization problems simultaneously, and this thesis has novelty in that it applied zero-shot learning techniques for the first time in this field, as presented in Figure 1.1.

The rest of this thesis is organized as follows. In chapter 2, we provide some background and related works. In chapter 3, we provide the learning-based approach to identifying types and positions of the sound sources and details of our audio datasets, SNU-B36-EX. Additionally, we verify that the datasets are classified in a supervised manner with modern deep models. In chapter 4, we train a generative model to learn a class-invariant mapping from seen classes and attempt to generate synthetic data of unseen classes which can be used to make our classifier applicable

to unseen data. In chapter 5, the zero-shot learning frameworks are applied for verifying the knowledge transferability from seen to unseen positions for several cases. Finally, we conclude our thesis in Chapter 6.

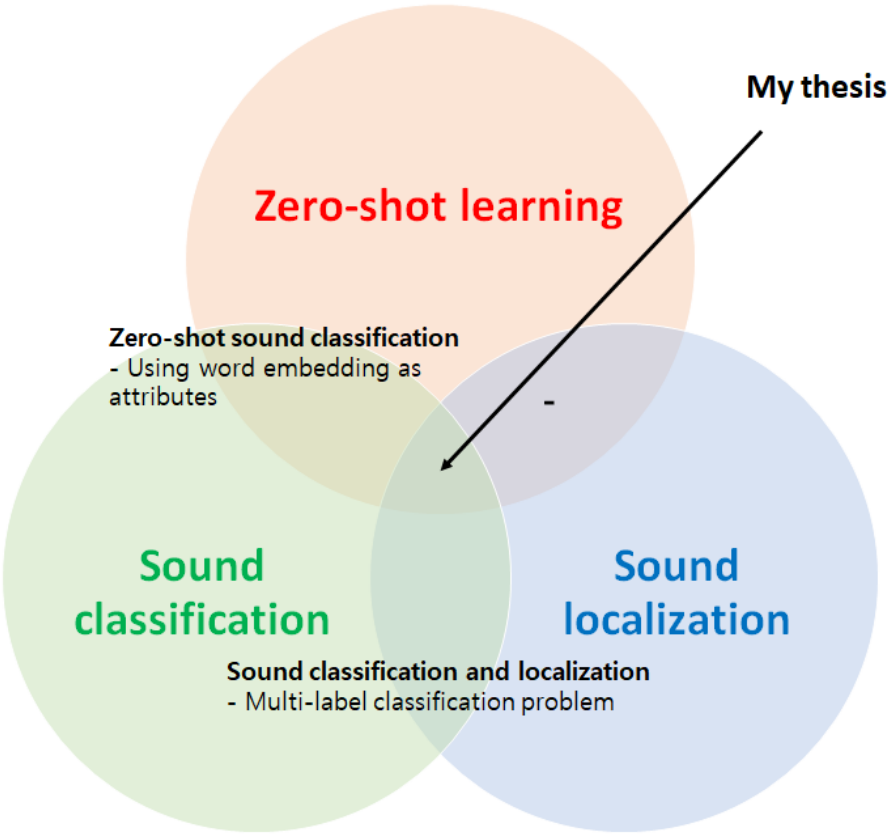


Figure 1.1: Scope of my thesis.

Bibliography

[1] P. Khunarsal, C. Lursinsap, and T. Raicharoen, "Very short time environmental sound classification based on spectrogram pattern matching," *Information Sciences*, vol. 243, pp. 57-74, 2013.

[2] A. J. Eronen, V. T. Peltonen, J. T. Tuomi, A. P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, "Audio-based context recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 321-329, 2005.

[3] W. Dargie, "Adaptive audio-based context recognition," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 39, no. 4, pp. 715-725, 2009.

[4] A. Rakotomamonjy, and G. Gasso, "Histogram of gradients of time-frequency representations for audio scene classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 142-153, 2014.

[5] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 151-155, 2015.

[6] V. Bisot, R. Serizel, S. Essid, and G. Richard, “Acoustic scene classification with matrix factorization for unsupervised feature learning,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6445-6449, 2016.

[7] J. F. Gemmeke, L. Vuegen, P. Karsmakers, B. Vanrumste, and H. Vanhamme, “An exemplar-based nmf approach to audio event detection,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustic*, pp. 1-4, 2013.

[8] J. C. Wang, J. F. Wang, K. W. He, and C. S. Hsu, “Environmental sound classification using hybrid svm/knn classifier and mpeg-7 audio lowlevel descriptor,” in *IEEE International Joint Conference on Neural Network*, pp. 1731-1735, 2006.

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

[10] K. J. Piczak, “Environmental sound classification with convolutional neural networks,” in *IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1-6, 2015.

[11] 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*, pp. 559-563, 2015.

[12] 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.

[13] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, “Acoustic scene classification: Classifying environments from the sounds they produce,” *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16-34, 2015.

[14] M. Valenti, A. Diment, G. Parascandolo, S. Squartini, and T. Virtanen, "DCASE 2016 acoustic scene classification using convolutional neural networks," in *Proc. Workshop Detection Classif. Acoust. Scenes Events*, pp.95-99, 2016.

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

[16] J. Sang, S. Park, and J. Lee, "Convolutional recurrent neural networks for urban sound classification using raw waveforms," in *IEEE European Signal Processing Conference*, pp.2444-2448, 2018.

[17] E. Cakır, 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*, vol. 25, no. 6, pp.1291-1303, 2017.

[18] S. Adavanne, P. Pertila, and T. Virtanen, "Sound event detection using spatial features and convolutional recurrent neural network," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp.771-775, 2017.

[19] H. Atmoko, D. Tan, G. Tian, and B. Fazenda, "Accurate sound source localization in a reverberant environment using multiple acoustic sensors," *Measurement Science and Technology*, vol. 19, no. 2, 2008.

[20] X. Alameda-Pineda, and R. Horaud, "A geometric approach to sound source localization from time-delay estimates," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 6, pp. 1082-1095, 2014.

[21] T. Gustafsson, B. D. Rao, and M. Trivedi, "Source localization in reverberant environments: Modeling and statistical analysis," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 791-803, 2003.

[22] F. Ribeiro, C. Zhang, D. A. Florencio, and D. E. Ba, "Using reverberation to improve range and elevation discrimination for small array sound source localization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1781-1792, 2010.

[23] I. An, D. Lee, J. Choi, D. Manocha, and S. Yoon, "Diffraction-aware sound localization for a non-line-of-sight source," in *IEEE International Conference on Robotics and Automation*, pp. 4061-4067, 2019.

[24] X. Sheng, and Y. H. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 53, no. 1, pp. 44-53, 2004.

[25] F. Deng, S. Guan, X. Yue, X. Gu, J. Chen, J. Lv, and J. Li, "Energy-based sound source localization with low power consumption in wireless sensor networks," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 4894-4902, 2017.

[26] A. Saxena, and A. Y. Ng, "Learning sound location from a single microphone," in *IEEE International Conference on Robotics and Automation*, pp. 1737-1742, 2009.

[27] R. Talmon, I. Cohen, and S. Gannot, "Supervised source localization using diffusion kernels," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 245-248, 2011.

[28] R. Parhizkar, I. Dokmanic, and M. Vetterli, "Single-channel indoor microphone localization," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1434-1438, 2014.

[29] T. T. Takashima, Ryoichi, and Y. Ariki, "HMM-based separation of acoustic transfer function for single-channel sound source localization," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2830-2833, 2010.

[30] S. Chakrabarty, and E. A. P. Habets, "Multi-Speaker DOA Estimation Using Deep Convolutional Networks Trained With Noise Signals," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 8-21, 2019.

[31] T. N. T. Nguyen, W. S. Gan, R. Ranjan, and D. L. Jones, "Robust Source Counting and DOA Estimation Using Spatial Pseudo-Spectrum and Convolutional Neural Network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2626-2637, 2020.

[32] J. Pak, and J. W. Shin, "Sound localization based on phase difference enhancement using deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1335-1345, 2019.

[33] R. Takeda, and K. Komatani, "Sound source localization based on deep neural networks with directional activate function exploiting phase information," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 405-409, 2016.

[34] R. Takeda, Y. Kudo, K. Takashima, Y. Kitamura, and K. Komatani, "Unsupervised adaptation of neural networks for discriminative sound source localization with eliminative constraint," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3514-3518, 2018.

[35] L. Perotin, R. Serizel, E. Vincent, and A. Guerin, "CRNN-Based Multiple DoA Estimation Using Acoustic Intensity Features for Ambisonics Recordings," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 22-33, 2019.

[36] W. He, P. Motlicek, and J. M. Odobez, "Adaptation of multiple sound source localization neural networks with weak supervision and domain-adversarial training," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 770-774, 2019.

[37] A. Politis, A. Mesaros, S. Adavanne, T. Heittola, and T. Virtanen, "Overview and Evaluation of Sound Event Localization and Detection in DCASE 2019," *arXiv preprint arXiv:2009.02792*, 2020.

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

[39] T. Butko, F. G. Pla, C. Segura, C. Nadeu, and J. Hernando, "Two-source acoustic event detection and localization: Online implementation in a Smart-room," in *European Signal Processing Conference*, pp. 1317-1321, 2011.

[40] K. Lopatka, J. Kotus, and A. Czyzewski, "Detection, classification and localization of acoustic events in the presence of background noise for acoustic surveillance of hazardous situations," *Multimedia Tools and Applications*, vol. 25, pp. 10407-10439, 2015.

[41] T. Hirvonen, "Classification of Spatial Audio Location and Content Using Convolutional Neural Networks," *Audio Engineering Society Convention*, vol. 2, 2015.

[42] K. Noh, C. Jeong-Hwan, J. Dongyeop, and C. Joon-Hyuk, "Three-stage approach for sound event localization and detection," in *Detection and Classification of Acoustic Scenes and Events Challenge*, 2019.

[43] R. Varzandeh, K. Adiloglu, S. Doclo, and V. Hohmann, "Exploiting Periodicity Features for Joint Detection and DOA Estimation of Speech Sources Using Convolutional Neural Networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 566-570, 2020.

[44] F. Ronchini, D. Arteaga, and A. Pérez-López, "Sound event localization and detection based on CRNN using rectangular filters and channel rotation data augmentation," *arXiv preprint arXiv:2010.06422*, 2020.

[45] 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, 2019.

[46] H. Phan, L. Pham, P. Koch, N. Q. Duong, I. McLoughlin, and A. Mertins, “On multitask loss function for audio event detection and localization,” *arXiv preprint arXiv:2009.05527*, 2020.

[47] T. Komatsu, M. Togami, and T. Takahashi, “Sound Event Localization and Detection Using Convolutional Recurrent Neural Networks and Gated Linear Units,” in *European Signal Processing Conference*, 2020.

[48] T. N. T. Nguyen, D. L. Jones, and W. S. Gan, “A Sequence Matching Network for Polyphonic Sound Event Localization and Detection,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 71-75, 2020.

[49] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, 2015.

[50] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing*, vol. 25, 2012.

[51] 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*, pp. 770–778, 2016.

[52] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, “Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, 2012.

[53] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.

[54] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *Advances in Neural Information Processing*, pp. 3111–3119, 2013.

[55] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Advances in Neural Information Processing*, pp. 3104–3112, 2014.

[56] H. Choi, H. Yang, S. Lee, and W. Seong, "Classification of Inter-Floor Noise Type/Position Via Convolutional Neural Network-Based Supervised Learning," *Applied Sciences*, vol. 9, no. 18, pp. 3735, 2019.

[57] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[58] Y. Fu, T. Xiang, Y. G. Jiang, X. Xue, L. Sigal, and S. Gong, "Recent advances in zero-shot recognition: Toward data-efficient understanding of visual content," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 112-125, 2018.

[59] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 453-465, 2013.

[60] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1425-1438, 2015.

[61] B. Romera-Paredes, and P. Torr, “An embarrassingly simple approach to zero-shot learning,” in *International Conference on Machine Learning*, pp. 2152-2161, 2015.

[62] L. Zhang, T. Xiang, and S. Gong, “Learning a Deep Embedding Model for Zero-Shot Learning,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[63] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, “Zero-shot learning through cross-modal transfer,” in *Advances in Neural Information Processing Systems Recognition*, pp. 935-943, 2013.

[64] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, “Devise: A deep visual-semantic embedding model,” in *Advances in Neural Information Processing Systems Recognition*, pp. 2121-2129, 2013.

[65] E. Kodirov, T. Xiang, and S. Gong, “Semantic autoencoder for zero-shot learning,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3174-3183, 2017.

Chapter 2

Backgrounds

2.1 Sound Classification

Sound classification, especially environmental sound classification, has been widely studied. In many studies, the conventional classification procedure is divided into two steps: feature extraction and classification. Well-known features extracted directly from raw audio include the zero-crossing rate,

$$\text{zero-crossing rate} = \frac{1}{T}\sum_{n=0}^{T-1} 1_{R_{<0}}\left(s(n)s(n-1)\right), \tag{2.1}$$

where s is a signal, T is length of the signal, and $1_{R_{<0}}$ is an indicator function, and time-averaged energy [2, 3],

$$\text{time-averaged energy} = \frac{1}{T}\sum_{n=0}^{T-1} \|s(n)\|^2. \tag{2.2}$$

The zero-crossing rate roughly estimates the dominant frequency of the signal and the time-averaged energy roughly indicates whether the signal of interest from background noise. Spectral features from the spectral domain related to frequencies of

the signal include the spectral centroid,

$$\text{spectral centroid} = C_i = \frac{\sum_{k=1}^{W} kX_i(k)}{\sum_{k=1}^{W} X_i(k)}, \tag{2.3}$$

where $X(k)$ is the weighted frequency coefficient of n-th bin, spectral spread,

$$\text{spectral spread} = \sqrt{\frac{\sum_{k=1}^{W}(k - C_i)^2 X_i(k)}{\sum_{k=1}^{W} X_i(k)}}, \tag{2.4}$$

which indicates the second central moment of the spectrum, and spectral flatness [8],

$$\text{spectral flatness} = \frac{\left(\prod_{k=1}^{N} X(k)\right)^{1/N}}{\frac{1}{N}\sum_{k=1}^{N} X(k)}. \tag{2.5}$$

Additionally, linear prediction coefficients are another popular feature values which are estimated as a linear combination of previous samples [1, 3],

$$\hat{x}(n) = \sum_{i=1}^{p} a_i x(n-i), \tag{2.6}$$

where $\hat{x}(n)$ is the predicted signal value, $x(n-i)$ is the previous observed samples, and a_i is the predictor coefficients. In recent years, Mel-frequency cepstral coefficients [1, 2, 3],

$$f_{mel} = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{2.7}$$

and the time-frequency representations [4, 5, 6, 7],

$$STFT\{x(n)\}(m,\omega) = \sum_{n=-\infty}^{\infty} x(n)w(n-m)\exp^{-j\omega n}, \tag{2.8}$$

where $w(n-m)$ is the window function, usually using a Hamming or Hann window, and $x(n)$ is the signal, have become the most used features applicable to the deep architectures.

Extracted features are classified using conventional machine learning methods, including dimension reduction methods, such as k-nearest neighbors algorithm [1, 2, 3, 8], hidden Markov models [2, 3], and matrix factorization [5, 6, 7],

$$\min_{W,H} \|V - WH\|_2^2 \qquad s.t. \quad W, H \geq 0. \tag{2.9}$$

and powerful traditional classifiers, such as support vector machine [4, 8],

$$\left[\frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i(w^T x_i - b))\right] + \lambda\|w\|^2. \tag{2.10}$$

However, in the recent era of deep learning, remarkable progress has been made using end-to-end learning, which enables the model to automatically learn representations of data, such as deep neural networks (DNNs) [9], convolutional neural networks (CNNs) [10, 11, 12, 13, 14], recurrent neural networks (RNNs) [15], and convolutional recurrent neural networks (CRNNs) [16, 17, 18]. As the sounds considered in this study propagate inside a building, our work is relevant to environmental sound classification. However, we would like to identify the position and type of sound.

2.2 Sound Source Localization

Sound source localization (SSL), especially for indoor sound or sound inside a building, has been extensively reported in the literature. Many existing studies on SSL obtain signals from multiple sensors or microphone arrays to utilize time/phase

differences [19, 20], reverberation [21, 22, 23], and energy-based information [24, 25]. Recently, there have been many attempts to solve SSL problem using learning-based methods in which the neural networks are trained with non-informative noise sources [30], diverse sound events [31, 32], speech of speakers [33, 34, 35], or simulated data [36]. However, they are not compatible with our problem, in which audio signals are recorded using a single microphone.

There have been few studies on SSL using a single microphone. In [26], a wide range of sounds was localized using a single microphone with artificial structures that mimic the outer ear of a human to modify the sound path depending on its incident angle. In [27], an algorithm was applied to recover the controlling parameter using diffusion kernels for SSL with a single microphone in a reverberant room. In [28], the image source method was applied to utilize room reverberation, which enables localization using a single microphone inside a known room. In [29], a voice was localized using a hidden Markov model to estimate the acoustic transfer function using a single microphone. The experiments were conducted in a known room or required a special setting to easily take advantage of reverberation or multipath effects. However, because the noise considered in this study comes from other rooms or floors, it does not necessarily accompany reverberation or multipath effects. In contrast to existing works, our approach considers the SSL problem as a position classification problem with a ZSL framework using a single microphone. Its position could be at any point in the building, even unseen during model training.

In recent years, the combined problem, called sound event localization and detection (SELD), has emerged in recent years. In SELD, spatio-temporal characterization of the acoustic scene is obtained by combining the sound source event detection (SED) and SSL [37]. Earlier studies on SELD separately treated the two problems with off-the-shelf machine learning methods for detection, and classic array processing methods for localization [38, 39, 40]. Recently, many attempts have been made to use neural networks as learning-based methods for SELD problems.

The joint probabilities for each type and position of the sound are predicted by a CNN or CRNN as a multi-label classification problem [41, 42, 43, 44]. Because the predicted output for positions is limited to the positions available in the training set, the performance for unseen positions is unknown. Meanwhile, in another direction of research, the predicted output consists of two branches: the output for the position is treated as a regression problem, while the output for types is treated as a classification problem [45, 46, 47, 48]. Our problem is similar to SELD, in that we are interested in both the identification of the type and position of sounds. However, in terms of the output format, the types of sound are classified per signal, not SED, which aims to detect the frame-wise occurrence of the sound events. In addition, while the existing SELD uses multi-channel signals given as the data set, only a single channel signal is given in our problem.

2.3 Supervised Learning

In supervised setting, train set is given as $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ are input and output pairs. The goal is training a model f_θ parameterized by θ, by solving

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, \mathcal{L}(\mathcal{D}; \theta), \tag{2.11}$$

where the $\mathcal{L}(\mathcal{D}; \theta)$ is a loss function that measures the error between predictions by the model θ and the true target values. There are two common loss functions that are used for supervised classification and regression with deep neural networks (DNN). For the supervised classification, the cross-entropy loss is usually used as

$$\mathcal{L}(\mathcal{D}; \theta) = \sum_{(x,y)\sim\mathcal{D}} \left(y \log f_\theta(x) + (1-y) \log \left(1 - f_\theta(x)\right)\right), \tag{2.12}$$

and for the supervised regression, the mean-squared error (MSE) is used as

$$\mathcal{L}(\mathcal{D};\theta) = \sum_{(x,y)\sim\mathcal{D}} \|y - f_\theta(x)\|_2^2 \,. \tag{2.13}$$

It is common sense that training a DNN requires a lot of data and expensive training procedures. Also, if there are few data available, it shows overfit on the train set and frequently loses their generalization ability on new tasks which is unseen during training.

2.4 Zero-Shot Learning

A goal of zero-shot learning (ZSL) is to train a model that can work well on the instances even unseen during training. In the ZSL setting, the target instance to be identified is totally unavailable during training which seems an almost impossible task to solve. The problem can be solved by what is inspired by the human ability to perceive new concepts from what one has previously seen and learned concepts [69]. An important assumption behind ZSL methods is that there is some shared structure of mapping between different modalities across classes. For example, for the image recognition domain which is one of the most active fields of ZSL, such modalities are visual feature-embeddings and the concepts of classes usually expressed as human-annotated attributes [55, 51, 52], text descriptions [70], or word-embeddings [54, 50]. The concepts of classes can be projected into some embedding space, called semantic-embeddings or class-embeddings. These semantic-embeddings can be used as side information to transfer knowledge from the seen classes to unseen classes in the ZSL setting. Then, the shared mapping model between feature-embeddings and semantic-embeddings across the seen classes during training can be expected to generalize on unseen classes, since the ZSL model aims

to capture the class-invariant mapping between those modalities.

ZSL has been widely studied in many fields. In the image domain, which is the most active field of ZSL, such modalities are represented as visual feature-embeddings and semantic-embeddings. From [49], the mapping function could be categorized by learning linear compatibility [50, 51, 52, 53], nonlinear compatibility [54], intermediate attribute classifiers [55], and hybrid models [72]. During testing, the learned model was evaluated to verify whether it could predict the unseen classes with the knowledge transferred through the semantic-embeddings.

ZSL has also been studied for character recognition [56], video classification [57], neural machine translation [58], and action recognition [59, 60]. Similar to our study, the ZSL framework has been applied to audio classification [61] and music classification problems [62]. In [61], audio feature-embeddings were extracted from a VGGish [63] model pre-trained with YouTube-8M and Word2Vec [64], which were used as class-embeddings. The mapping function was trained to maximize the linear compatibility between the two modalities, similar to that in [51]. The model is evaluated using the public audio dataset ESC-50 [65]. The researchers of the study [62] used the output of their CNN model, which considers audio Mel-spectrogram as input, as feature-embedding, and instrument attributes and GloVe [66] as class-embedding. Their mapping function was trained using max-margin hinge loss, which enforces a certain margin between the compatibility values from positive and negative sampling, similar to [50]. The model was evaluated on the free music archive (FMA) [67] and the million song dataset (MSD) [68]. These studies are similar to those of our study in the incorporation of audio data. However, they differ in terms of solely classifying the type of sound, while we focus on the type and position of the sound source.

For the standard ZSL problems, we are given train set $\mathcal{D}^s = \{(x_i^s, y_i^s) | x_i^s \in \mathcal{X}^s, y_i^s \in \mathcal{Y}^s\}_{i=1}^{N^s}$ where $x_i^s \in \mathcal{X}^s$ denotes an feature-embeddings from seen classes, and $y_i^s \in \mathcal{Y}^s$ denotes the corresponding label that is one of the seen classes $\mathcal{Y}^s$.

The attributes of seen classes are $\mathcal{A}^s = \{a(y)|y \in \mathcal{Y}^s\}$. Attribute $a(y)$ explains the semantic information of y as vector form. Using this side information, we can model the relationship between all classes, and transfer the knowledge from the seen to unseen classes. We are given classes that are assumed to be the unseen potential target classes. We are given sets of unseen classes $\mathcal{Y}^u$ and the corresponding unseen attributes $\mathcal{A}^u = \{a(y)|y \in \mathcal{Y}^u\}$. Here, the unseen classes are disjoint from the seen classes, i.e., $\mathcal{Y}^s \cap \mathcal{Y}^u = \emptyset$. Unlike a set of seen data S, we cannot access features from unseen classes, i.e., $x_i^u \in \mathcal{X}^u$ is not available during training, but is appear during the testing phase. Thus, we have class-level information on the unseen classes during the training phase. During the training, the following empirical risk

$$W^* = \underset{W}{\operatorname{argmin}}\, E_{(x,y)\sim P(\mathcal{X}^s,\mathcal{Y}^s)}\mathcal{L}(f(x;W),y), \tag{2.14}$$

is minimized by using the train set S, where $\mathcal{L}$, and f are the loss and mapping functions parameterized by the W, respectively. We expect that the parameters trained with data from seen classes further generalize well on unseen data from the unseen classes, $(x, y) \sim P(\mathcal{X}^u, \mathcal{Y}^u)$. The empirical risk to evaluate the trained parameter W for the ZSL setting can be defined as

$$E_{(x,y)\sim P(\mathcal{X}^u,\mathcal{Y}^u)}\mathcal{L}(f(x;W^*),y), \tag{2.15}$$

when the test set consists of data from only unseen classes. During testing, the learned model is evaluated to verify whether it could predict the unseen classes with the knowledge transferred through the semantic-embeddings. From [49], there have been several ways to build the mapping functions categorized by learning linear compatibility [51, 52, 53], neural networks [54], intermediate attribute classifiers [55], and hybrid models [72].

Although the ZSL setting assumes that the test examples are only from unseen

classes, considering real scenarios, test examples could arise from all classes, including seen and unseen classes. [49] established generalized zero-shot learning (GZSL) where the test examples can be from, and expected to be classified as, either seen or unseen classes. The empirical risk to evaluate the trained parameter W for the GZSL setting can be defined as

$$E_{(x,y)\sim P(\mathcal{X}^{s+u},\mathcal{Y}^{s+u})}\mathcal{L}(f(x;W^*),y), \tag{2.16}$$

when the test set consists of data from the seen and unseen classes. Here, $\mathcal{X}^{s+u} = \mathcal{X}^s \cup \mathcal{X}^u, \mathcal{Y}^{s+u} = \mathcal{Y}^s \cup \mathcal{Y}^u$. However, existing methods have been reported to be inefficient and easily susceptible under the GZSL setting [69, 49, 73], and efficient methods utilizing the generative models, such as variational autoencoders (VAEs) [74], or generative adversarial networks (GANs) [75] have drawn attention recently [73, 76, 77, 78, 79, 80].

From [49, 71], the approaches for zero-shot learning are usually categorized by projection-based methods [51, 52, 50] and generative model-based methods [73, 76, 77, 78, 79, 80].

First, the projection-based methods aim to learn the compatibility function between the pairs of embeddings of different modalities is measured the correspondence between them [51, 52, 50, 53]. The compatibility function is defined as,

$$F(x,y;W) = \theta(x)^T W \phi(y), \tag{2.17}$$

where $\theta(x)$ and $\phi(y)$ are feature-embeddings and semantic-embeddings, and W is the learnable mapping function between those embeddings. From [51, 52, 50], the pairwise ranking objective is defined to minimized,

$$\sum_{y \in \mathcal{Y}^{tr}} [\Delta(y_n, y) + F(x_n, y; W) - F(x_n, y_n; W)], \tag{2.18}$$

where $\Delta(y_n, y)$ is equal to 1 if $y_n = y$, otherwise 0. [53] also learns the projection from feature-embeddings to semantic-embeddings, but it further constrains that the projection must be able to reconstruct the original embeddings. The objective to minimize can be defined,

$$\min_W \|\theta(x) - W^T \phi(y)\|^2 + \lambda \|W\theta(x) - \phi(y)\|^2, \tag{2.19}$$

where λ is a hyperparameter to be balanced between two reconstruction losses.

However, existing projection-based methods have been reported to be inefficient and easily susceptible under GZSL setting [69, 49, 73], and efficient methods utilizing the generative models, such as variational autoencoders (VAEs) [74], or generative adversarial networks (GANs) [75]. The generative-based methods aim to learn to obtain instances for the unseen classes by synthesizing some pseudo instances [71]. Starting from the empirical risk of the classifier,

$$E_{(x,y) \sim P(\mathcal{X}^s, \mathcal{Y}^s)} \mathcal{L}(f(x; W), y), \tag{2.20}$$

was minimized by training the training set S, where $\mathcal{L}$, and f are the loss and mapping functions parameterized by W, respectively. During the training phase, data from the seen classes can be used. We expect that the parameters from the aforementioned training process further minimize the following risks. The risks are defined through the trainset domain, for the ZSL setting as,

$$E_{(x,y) \sim P(\mathcal{X}^u, \mathcal{Y}^u)} \mathcal{L}(f(x; W), y), \tag{2.21}$$

when the test set consists of data from only unseen classes, and for the GZSL setting,

$$E_{(x,y)\sim P(\mathcal{X}^{s+u},\mathcal{Y}^{s+u})}\mathcal{L}(f(x;W),y), \tag{2.22}$$

[73] uses the GANs as the generative model consisting of a conditional generator $G : \mathcal{Z} \times \mathcal{C} \rightarrow \mathcal{X}$ parameterized by θ_G, and a conditional discriminator $D : \mathcal{X} \times \mathcal{C} \rightarrow [0, 1]$ parameterized by θ_D. Here, the generator G takes a random Gaussian noise vector $z \in \mathcal{Z} \sim \mathcal{N}(0, 1)$ and class-embedding $c(y) \in \mathcal{C}$, and yields fake features $\tilde{x}$ corresponding to class y. The simple version of the loss is as follows:

$$\mathcal{L}_{GAN}(\theta_G, \theta_D) = E[\log D(x, c(y_s); \theta_D)] + E[\log(1 - D(\tilde{x}, c(y_s); \theta_D))], \tag{2.23}$$

where $\tilde{x} = G(z, c(y_s); \theta_G)$ denotes fake features corresponding to the seen class-embeddings $c(y_s)$.

$$\theta_G^*, \theta_D^* = \arg\min_{\theta_G} \max_{\theta_D} L_{GAN}(\theta_G, \theta_D), \tag{2.24}$$

where θ_G^* and θ_D^* are the optimal parameters for the generator and discriminator of the model trained by the seen classes, respectively; and β is a hyperparameter of the classifier loss weight. The generative model can be replaced by every promising generative model, such as VAEs [76, 77].

After training the generative model, we synthesize the same number (n) of unseen features for each unseen class using their class-embeddings, $\tilde{\mathcal{D}}^u = \{(\tilde{x}_i^u, y_i^u, a(y_i^u)) | \tilde{x}_i^u \in \mathcal{X}^{\tilde{u}}, y_i^u \in \mathcal{Y}^u, a(y_i^u) \in \mathcal{A}^u\}_{i=1}^{N^u}$ where $\tilde{x}_i^u = G(z, c(y_i^u); \theta_G^*)$ and $\mathcal{X}^{\tilde{u}}$ denote the synthesized feature distribution from unseen classes. By training the new training set, the following modified empirical risk can be minimized for the ZSL

setting:

$$E_{(x,y)\sim P(\mathcal{X}^{\tilde{u}},\mathcal{Y}^{u})}\mathcal{L}(f(x;W),y), \tag{2.25}$$

which denotes that the classifier f parametrized by W can be trained with the synthesized features from unseen classes. For the GZSL setting, the synthesized features can be combined with features from the seen classes as a new training set. The combined set is utilized to train the classifier, f, to minimize the empirical risk for the GZSL setting:

$$E_{(x,y)\sim P(\mathcal{X}^{s+\tilde{u}},\mathcal{Y}^{s+u})}\mathcal{L}(f(x;W),y), \tag{2.26}$$

where a feature can be from either seen features x_s or synthesized features $\tilde{x}^u$ from unseen classes, that is, $\mathcal{X}^{s+\tilde{u}}$, and the corresponding class is one of all the classes, including the seen and unseen classes $\mathcal{Y}^{s+u}$.

Bibliography

[1] P. Khunarsal, C. Lursinsap, and T. Raicharoen, "Very short time environmental sound classification based on spectrogram pattern matching," *Information Sciences*, vol. 243, pp. 57-74, 2013.

[2] A. J. Eronen, V. T. Peltonen, J. T. Tuomi, A. P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, "Audio-based context recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 321-329, 2005.

[3] W. Dargie, "Adaptive audio-based context recognition," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 39, no. 4, pp. 715-725, 2009.

[4] A. Rakotomamonjy, and G. Gasso, "Histogram of gradients of time-frequency representations for audio scene classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 142-153, 2014.

[5] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 151-155, 2015.

[6] V. Bisot, R. Serizel, S. Essid, and G. Richard, “Acoustic scene classification with matrix factorization for unsupervised feature learning,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6445-6449, 2016.

[7] J. F. Gemmeke, L. Vuegen, P. Karsmakers, B. Vanrumste, and H. Vanhamme, “An exemplar-based nmf approach to audio event detection,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustic*, pp. 1-4, 2013.

[8] J. C. Wang, J. F. Wang, K. W. He, and C. S. Hsu, “Environmental sound classification using hybrid svm/knn classifier and mpeg-7 audio lowlevel descriptor,” in *IEEE International Joint Conference on Neural Network*, pp. 1731-1735, 2006.

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

[10] K. J. Piczak, “Environmental sound classification with convolutional neural networks,” in *IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1-6, 2015.

[11] 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*, pp. 559-563, 2015.

[12] 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.

[13] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, “Acoustic scene classification: Classifying environments from the sounds they produce,” *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16-34, 2015.

[14] M. Valenti, A. Diment, G. Parascandolo, S. Squartini, and T. Virtanen, "DCASE 2016 acoustic scene classification using convolutional neural networks," in *Proc. Workshop Detection Classif. Acoust. Scenes Events*, pp.95-99, 2016.

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

[16] J. Sang, S. Park, and J. Lee, "Convolutional recurrent neural networks for urban sound classification using raw waveforms," in *IEEE European Signal Processing Conference*, pp.2444-2448, 2018.

[17] E. Cakır, 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*, vol. 25, no. 6, pp.1291-1303, 2017.

[18] S. Adavanne, P. Pertila, and T. Virtanen, "Sound event detection using spatial features and convolutional recurrent neural network," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp.771-775, 2017.

[19] H. Atmoko, D. Tan, G. Tian, and B. Fazenda, "Accurate sound source localization in a reverberant environment using multiple acoustic sensors," *Measurement Science and Technology*, vol. 19, no. 2, 2008.

[20] X. Alameda-Pineda, and R. Horaud, "A geometric approach to sound source localization from time-delay estimates," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 6, pp. 1082-1095, 2014.

[21] T. Gustafsson, B. D. Rao, and M. Trivedi, "Source localization in reverberant environments: Modeling and statistical analysis," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 791-803, 2003.

[22] F. Ribeiro, C. Zhang, D. A. Florencio, and D. E. Ba, “Using reverberation to improve range and elevation discrimination for small array sound source localization,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1781-1792, 2010.

[23] I. An, D. Lee, J. Choi, D. Manocha, and S. Yoon, “Diffraction-aware sound localization for a non-line-of-sight source,” in *IEEE International Conference on Robotics and Automation*, pp. 4061-4067, 2019.

[24] X. Sheng, and Y. H. Hu, “Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks,” *IEEE Transactions on Signal Processing*, vol. 53, no. 1, pp. 44-53, 2004.

[25] F. Deng, S. Guan, X. Yue, X. Gu, J. Chen, J. Lv, and J. Li, “Energy-based sound source localization with low power consumption in wireless sensor networks,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 4894-4902, 2017.

[26] A. Saxena, and A. Y. Ng, “Learning sound location from a single microphone,” in *IEEE International Conference on Robotics and Automation*, pp. 1737-1742, 2009.

[27] R. Talmon, I. Cohen, and S. Gannot, “Supervised source localization using diffusion kernels,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 245-248, 2011.

[28] R. Parhizkar, I. Dokmanic, and M. Vetterli, “Single-channel indoor microphone localization,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1434-1438, 2014.

[29] T. T. Takashima, Ryoichi, and Y. Ariki, “HMM-based separation of acoustic transfer function for single-channel sound source localization,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2830-2833, 2010.

[30] S. Chakrabarty, and E. A. P. Habets, "Multi-Speaker DOA Estimation Using Deep Convolutional Networks Trained With Noise Signals," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 8-21, 2019.

[31] T. N. T. Nguyen, W. S. Gan, R. Ranjan, and D. L. Jones, "Robust Source Counting and DOA Estimation Using Spatial Pseudo-Spectrum and Convolutional Neural Network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2626-2637, 2020.

[32] J. Pak, and J. W. Shin, "Sound localization based on phase difference enhancement using deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1335-1345, 2019.

[33] R. Takeda, and K. Komatani, "Sound source localization based on deep neural networks with directional activate function exploiting phase information," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 405-409, 2016.

[34] R. Takeda, Y. Kudo, K. Takashima, Y. Kitamura, and K. Komatani, "Unsupervised adaptation of neural networks for discriminative sound source localization with eliminative constraint," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3514-3518, 2018.

[35] L. Perotin, R. Serizel, E. Vincent, and A. Guerin, "CRNN-Based Multiple DoA Estimation Using Acoustic Intensity Features for Ambisonics Recordings," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 22-33, 2019.

[36] W. He, P. Motlicek, and J. M. Odobez, "Adaptation of multiple sound source localization neural networks with weak supervision and domain-adversarial training," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 770-774, 2019.

[37] A. Politis, A. Mesaros, S. Adavanne, T. Heittola, and T. Virtanen, "Overview and Evaluation of Sound Event Localization and Detection in DCASE 2019," *arXiv preprint arXiv:2009.02792*, 2020.

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

[39] T. Butko, F. G. Pla, C. Segura, C. Nadeu, and J. Hernando, "Two-source acoustic event detection and localization: Online implementation in a Smart-room," in *European Signal Processing Conference*, pp. 1317-1321, 2011.

[40] K. Lopatka, J. Kotus, and A. Czyzewski, "Detection, classification and localization of acoustic events in the presence of background noise for acoustic surveillance of hazardous situations," *Multimedia Tools and Applications*, vol. 25, pp. 10407-10439, 2015.

[41] T. Hirvonen, "Classification of Spatial Audio Location and Content Using Convolutional Neural Networks," *Audio Engineering Society Convention*, vol. 2, 2015.

[42] K. Noh, C. Jeong-Hwan, J. Dongyeop, and C. Joon-Hyuk, "Three-stage approach for sound event localization and detection," in *Detection and Classification of Acoustic Scenes and Events Challenge*, 2019.

[43] R. Varzandeh, K. Adiloglu, S. Doclo, and V. Hohmann, "Exploiting Periodicity Features for Joint Detection and DOA Estimation of Speech Sources Using Convolutional Neural Networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 566-570, 2020.

[44] F. Ronchini, D. Arteaga, and A. Pérez-López, "Sound event localization and detection based on CRNN using rectangular filters and channel rotation data augmentation," *arXiv preprint arXiv:2010.06422*, 2020.

[45] 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, 2019.

[46] H. Phan, L. Pham, P. Koch, N. Q. Duong, I. McLoughlin, and A. Mertins, "On multitask loss function for audio event detection and localization," *arXiv preprint arXiv:2009.05527*, 2020.

[47] T. Komatsu, M. Togami, and T. Takahashi, "Sound Event Localization and Detection Using Convolutional Recurrent Neural Networks and Gated Linear Units," in *European Signal Processing Conference*, 2020.

[48] T. N. T. Nguyen, D. L. Jones, and W. S. Gan, "A Sequence Matching Network for Polyphonic Sound Event Localization and Detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 71-75, 2020.

[49] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning-A comprehensive evaluation of the good, the bad and the ugly," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2251-2265, 2018.

[50] A. Frome, G. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, "DeViSE: A Deep Visual-Semantic Embedding Model," in *Advances in Neural Information Processing Systems*, 2013.

[51] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1425-1438, 2015.

[52] B. Romera-Paredes, and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *International Conference on Machine Learning*, pp. 2152-2161, 2015.

[53] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3174-3183, 2017.

[54] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," in *Advances in Neural Information Processing Systems Recognition*, pp. 935-943, 2013.

[55] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 453-465, 2013.

[56] H. Larochelle, D. Erhan, and Y. Bengio, "Zero-data learning of new tasks," in *Association for the Advancement of Artificial Intelligence*, vol. 1, no. 2, 2008.

[57] J. Gao and C. Xu, "CI-GNN: Building a Category-Instance Graph for Zero-Shot Video Classification," *IEEE Transactions on Multimedia*, 2020.

[58] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viegas, M. Wattenberg, and G. Corrado, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 339-351, MIT Press, 2017.

[59] X. Xu, T. Hospedales, and S. Gong, "Transductive zero-shot action recognition by word-vector embedding," *International Journal of Computer Vision*, vol. 123, no. 3, pp. 309-333, Springer, 2017.

[60] J. Qin, L. Liu, L. Shao, F. Shen, B. Ni, J. Chen, and Y. Wang, "Zero-shot action recognition with error-correcting output codes," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2833-2842, 2017.

[61] H. Xie and T. Virtanen, "Zero-Shot Audio Classification Based On Class Label Embeddings," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 264-267, 2019.

[62] J. Choi, J. Lee, J. Park, and J. Nam, "Zero-shot learning for audio-based music classification and tagging," *arXiv preprint arXiv:1907.02670*, 2019.

[63] S. Hershey, S. Chaudhuri, D. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," in *International Conference on Acoustics, Speech and Signal Processing*, pp. 131-135, 2017.

[64] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, pp. 3111-3119, 2013.

[65] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Proceedings of the ACM International Conference on Multimedia*, pp. 1015-1018, 2015.

[66] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1532-1543, 2014.

[67] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "Fma: A dataset for music analysis," *arXiv preprint arXiv:1612.01840*, 2016.

[68] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," *arXiv preprint arXiv:1612.01840*, 2011.

[69] Y. Fu, T. Xiang, Y. G. Jiang, X. Xue, L. Sigal, and S. Gong, "Recent advances in zero-shot recognition: Toward data-efficient understanding of visual content," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 112-125, 2018.

[70] L. Zhang, T. Xiang, and S. Gong, "Learning a deep embedding model for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[71] W. Wang, V. W. Zheng, H. Yu, and C. Miao, "A Survey of Zero-Shot Learning: Settings, Methods, and Applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, 2019.

[72] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean, "Zero-shot learning by convex combination of semantic embeddings," in *International Conference on Learning Representations*, 2014.

[73] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5542–5551, 2018.

[74] D. P. Kingma, and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations*, 2014.

[75] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

[76] V. Kumar Verma, G. Arora, A. Mishra, and P. Rai, "Generalized zero-shot learning via synthesized examples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4281–4289, 2018.

[77] E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata, "Generalized zero-and few-shot learning via aligned variational autoencoders," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8247–8255, 2019.

[78] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, and A. Elgammal, "A generative adversarial approach for zero-shot learning from noisy texts," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1004–1013, 2018.

[79] R. Felix, V. B. Kumar, I. Reid, and G. Carneiro, "Multi-modal cycle-consistent generalized zero-shot learning," in *Proceedings of the European Conference on Computer Vision*, pp. 21–37, 2018.

[80] Y. Xian, S. Sharma, B. Schiele, and Z. Akata, "f-vaegan-d2: A feature generating framework for any-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Chapter 3

Identifying Type and Position of Sound Source as Supervised Classification Problem

3.1 Introduction

3.1.1 Motivation

Sounds caused by people working in an office building or living in an apartment such as footsteps, hammering, dragging of furniture, and noise from electrical appliances can be annoying. These sounds propagate along with the building structure, including walls, floors, ceilings, and columns, resulting in the propagation of this noise to all residents. It is often difficult to identify the source of the sound in such a real building, and successfully identifying the annoying sounds by their types and positions is necessary to mitigate the ensuing problem [1, 2, 3, 4, 5].

3.1.2 Related works

Clearly classifying such sounds by type and position can be one way to solve conflicts. Classifying those sounds could be a subset of the acoustic scene classification (ASC) problem. Remarkable progress has been made in recent years on ASC problems with a wide variety of signal processing and machine learning techniques.

There are papers including hidden Markov models [6], support vector machines [7], non-negative matrix factorization [8, 9, 10], deep neural networks [11], and convolutional neural networks [12, 13, 14]. For those audio classification tasks, each recorded sound is usually transformed into proper time-frequency representation which has become normal in many related tasks, recently [8, 9, 10, 11, 12, 13, 14].

3.1.3 Contributions of this chapter

In this chapter, we propose a learning-based approach to identify the type and position of sounds using a single microphone in a real-world building and verify that our audio datasets are well classified with modern deep architectures. We attempt to treat this problem as a joint classification problem in which each type and position is jointly classified in a supervised manner. We use two modules for the joint classification problem where one is convolutional neural networks (CNN) trained from scratch on our training dataset and the other is pre-trained CNN models trained with a large open-sourced audio dataset [15]. While there are two differences between ASC and our problem. ASC focuses solely on the type of a sound, our classification problem not only classifies the type of a sound but also tries to classify the position of the sound. We expect that even data of the same type of sound can be classified depending on their different positions by our classification models.

Additionally, unlike ASC data [16, 17, 18], a large noise dataset for classification does not exist and it is hard to obtain a large amount of the data especially when people are living in the building. In order to construct our dataset, SNU-B36-EX [19], using a single microphone, we collected 8,450 audio events that are generated from 5 pre-defined source types at nearly 39 different positions in the building, i.e., a classification problem with 169 classes. Then, we evaluate our deep architectures on the dataset and verify that our deep architectures are sufficient to classify the audio events.

3.2 Approach

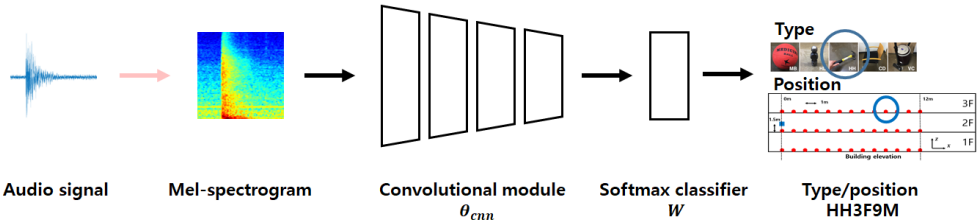


Figure 3.1: Overall procedures for identifying the type and position of the sound source.

The overall procedures for identifying the type and position of the sound source are as follows: (1) pre-processing: recorded audio signal data x is transformed into time-frequency representation such as Mel-spectrograms, (2) training: the deep architecture followed by softmax classifier is trained in an end-to-end manner with the given train set, and (3) testing: the trained model is evaluated on test set where the class of audio signal is predicted by the model. The overall procedures are described in Figure 3.1.

3.2.1 Pre-processing

In pre-processing, the recorded audio signal data x is transformed into time-frequency representation. First, the signal is sliced to an exact size long including a fixed number of samples, and normalized by the maximum value of each signal. Second, the signal is transformed by a short-time Fourier transform, and the transformed value with frequency ω at m^{th} time bin is

$$STFT\{x(n)\}(m,\omega) = \sum_{n=-\infty}^{\infty} x(n)w(n-m)\exp^{-j\omega n}, \tag{3.1}$$

where $w(n-m)$ is the window function, usually using a Hamming or Hann window. Then, the Mel-filter bank is computed as a weighted matrix looking at the spectrum

fine at frequencies sensitive to human auditory and coarse at the rest of the frequencies,

$$
\begin{aligned}
m &= 2595log_{10}(1 + \frac{f}{700}), \\
f &= 700(10^{\frac{m}{2595}} - 1),
\end{aligned}
\tag{3.2}
$$

$$
MB_m(\omega) = \begin{cases}
0 & \omega < f(m-1) \\
\frac{\omega - f(m-1)}{f(m) - f(m-1)} & f(m-1) \leq \omega < f(m) \\
1 & \omega = f(m) \\
\frac{f(m+1) - \omega}{f(m+1) - f(m)} & f(m) < \omega \leq f(m+1) \\
0 & \omega > f(m+1)
\end{cases}
\tag{3.3}
$$

Finally, the Mel-spectrogram is the multiplication of the Mel-filter bank and the power-spectrogram, which is squared of the absolute value of the time-frequency representation.

3.2.2 Training and testing

Our given training set $D_{tr} = \{(x_i, y_i) | x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}_{i=1}^{N}$ where $x_i \in \mathcal{X}$ denotes an audio signal data, and $y_i \in \mathcal{Y}$ denotes the corresponding label. M being the number of classes belong to $\mathcal{Y}$. During the training phase, a convolutional neural networks $\Theta = \{\theta_{cnn}, W\}$, including convolutional module θ_{cnn} and the common softmax classifier W minimizes the objective,

$$
\theta^*_{cnn}, W^* = \underset{\Theta}{\operatorname{argmin}} -E_{(x,y) \in D_{tr}} \left[logP(y|x; \Theta) \right], \tag{3.4}
$$

where $\Theta = \{\theta_{cnn}, W\}$ are updated simultaneously during the training where $W \in R^{d_x \times M}$ is the classifier weight matrix usually being a fully connected layer, and the

probabilities of class y are

$$P(y|x;\Theta) = \frac{\exp(W_{:,y}^T f(x;\theta_{cnn}))}{\sum_{i=1}^{N} \exp(W_{:,i}^T f(x;\theta_{cnn}))}, \tag{3.5}$$

where $W_{:,i}^T$ denotes the transpose of the ith column of the matrix W, and $f(x;\theta_{cnn})$ is output representation of x through the convolutional module θ_{cnn}. During the testing phase, the predicted class of x is

$$\underset{y}{\operatorname{argmax}}\, P(y|x;\theta_{cnn}^*, W). \tag{3.6}$$

3.3 Dataset Construction

Figure 3.2: Five types of sound sources include dropping a medicine ball on the floor (MB) for footstep sounds, dropping a hammer on the floor (HD), hitting with a hammer on the floor (HH) for hammering sounds, dragging a chair on the floor (CD) for sounds of dragging furniture, and operating a vacuum cleaner (VC) for electrical appliances [5].

The dataset SNU-B36-EX is an extended version of the inter-floor noise dataset (SNU-B36-50) from a previous study [5]. SNU-B36-EX consists of five types of sound sources to emulate most complaints reported by the Floor Noise Management Center [4]. These types were generated by dropping a medicine ball on the floor (MB) for footstep sounds, dropping a hammer on the floor (HD), hitting with a hammer on the floor (HH) for hammering sounds, dragging a chair on the floor

(CD) for sounds of dragging furniture, and operating a vacuum cleaner (VC) for electrical appliances as described in Figure 3.2. Each sound was from one of the 39 positions, as presented in Figure 3.3. The positions of the recorder and sound source are represented by blue squares and red circles, respectively. Each point is on one of the 1st, 2nd, and 3rd floors with a horizontal (x-axis) distance of 0 - 12 m at 1 m intervals from the recorder. The recorder was positioned at a height of 1.5 m on the 2nd floor and 4 m away from any source in the z-axis direction. We acquired 50 recordings for each class that contained one combination of each type and position.

SNU-B36-EX was collected from Bldg. 36 at Seoul National University. The sounds were recorded by a single smartphone at a sampling rate of 44.1 kHz. We sliced each audio signal to equal lengths of 2.4 s, which sufficiently contained the sound of the corresponding class. The VC for the 1st and 3rd floor data were excluded, as there was no significant signal recorded by the VC in the 1st and 3rd floor data. In this study, we define each class as (sound type)(position value in the x-axis)(floor). For example, a class called HD6M3F indicates that it contains sounds of dropping a hammer from 6 m on the 3rd floor. In summary, there are 8450 recordings categorized by 169 different labels.

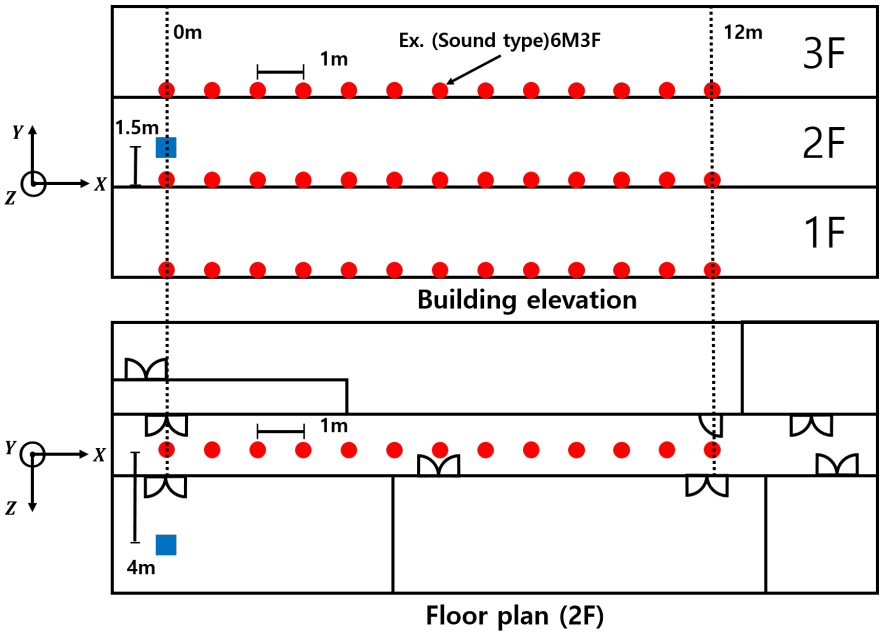


Figure 3.3: Bldg. 36 at Seoul National University viewed from the side (above) and from the top (bottom) with positions of the sound source (red circles) and receiver (blue square) [19]. X, Y, and Z indicate the axes and are not related to the features and class labels, respectively. For example, the class indicated by the arrow in the upper figure is sound type 6M3F.

3.4 Experiments

3.4.1 Experimental settings

We used two modules to extract audio feature-embeddings, VGGish [15] and a one-dimensional CNN (1D-CNN), pre-trained on our training set. VGGish is a pre-trained CNN model that uses the 'YouTube-8M' dataset [15]. Following [15], each audio dataset is transformed into a log-Mel-spectrogram with 64 frequency bins with a sampling rate of 16 kHz. As our audio data is 2.4 s long, the transformed spectrogram has a size of 64×242. The model has a fixed-size input of 64×96 and results in 128-dimension features. Because of the input size discrepancy, we extract features through VGGish with a hop length of 23 frames and take the element-wise average over them. As the module does not have access to the training set, it is used as a task-general feature extractor to prevent the feature-embeddings from being excessively biased toward our dataset.

1D-CNN is used as a task-specific feature extractor trained using our training set. All audio data were transformed into a Mel-spectrogram with 120 frequency bins. We used a pre-emphasis [20] of 0.95, a Hamming window with a length of 2205, and a hop length of 441 for the transformation. After transforming into a Mel-spectrogram, we trained the 1D-CNN with the training set at each data split scheme without any pre-training, which makes the extractor task-specific. The architecture of the 1D-CNN consists of two consecutive blocks with two convolutional layers, an average pooling layer, and two fully connected layers. Batch normalization [21] is located after each convolutional layer. The widths of the convolutional filters were 7, 5, 7, and 5, and the strides were all 1. We used zero-padding to match the lengths of the inputs and outputs for the convolutional layers. The widths and strides of the pooling layers were 3. The number of filters at each convolutional layer and hidden unit in the fully connected layers was 128. All the activation functions are ELU [22]. All hyperparameters were selected by trial and error, and early stopping was used

to obtain the best performance. Feature representations can be extracted using either VGGish, 1D-CNN models, or both. Using only VGGish or 1D-CNN, the output feature has 128-dimensions. When using both extractors, the output features are concatenated and yield a 256-dimensional vector. The features are standardized with zero mean and unit variance.

3.4.2 Experimental results

Table 3.1: Comparison of classification performance (%) for the feature representations under supervised learning.

Feature extractor	VGGish[15]	1D-CNN	VGGish [15] + 1D-CNN
Pre-trained on trainset	No	Yes	No+Yes
Dimension	128	128	256
Classification accuracy (%)	74.15	92.96	**96.96**

Although the SNU-B36-EX dataset was originally collected for zero-shot related tasks, we compared the representations under the supervised settings. Table 3.1 compares the classification performance of the feature representations under the supervised learning settings. Using both feature representations was the best choice in all settings. The best performance is obtained by two combined modules as 96.96% which can be considered that our datasets can be sufficient to classify by the 256-dimensional vector representations learned from the end-to-end learning procedures.

Comparing the individual feature representations, as the 1D-CNN is trained on the training set while the VGGish is not, the classification performance on the 1D-CNN features is 92.96%, which is better than that of 74.15% of the VGGish features under the supervised learning setting. VGGish is pre-trained with a much larger dataset, which ensures generalizability, whereas the 1D-CNN is more biased toward

the training set. These two extractors could be complementary to each other to extract meaningful features from our audio dataset. Therefore, we use the combined feature representations for experiments in further chapters.

3.5 Conclusion

For identifying the type and position of indoor noise, we collected the indoor noise data from a building and labeled it with its type and position simultaneously. Then, we transformed the recordings into Mel-spectrograms utilizing several well-known pre-processing techniques. After pre-processing, we exploit two modules which are 1D-CNN from scratch as a task-specific feature extractor and pre-trained vggish [15] as a task-general feature extractor. For selecting the best feature extractor for our classification problem and further zero-shot problems, we evaluate their classification performance of them. The best performance is obtained when using both modules, resulting in top-1 accuracy of 96.96 %. In the later chapter, we would apply the feature extractor pre-trained with the corresponding given train set. Also, we intend to see whether we could classify the type and position of indoor noise from other situations such as some part of the data is not available during the training.

Bibliography

[1] J. Y. Jeon, J. K. Ryu, and P. J. Lee, "A quantification model of overall dissatisfaction with indoor noise environment in residential buildings," *Applied Acoustics*, vol. 71, pp. 914-921, 2010.

[2] J. Ryu, H. Sato, K. Kurakata, A. Hiramitsu, M. Tanaka, and T. Hirota, "Relation between annoyance and single-number quantities for rating heavy-weight floor impact sound insulation in wooden houses," *Journal of the Acoustical Society of America*, vol. 129, pp. 3047-3055, 2011.

[3] S. H. Park, P. J. Lee, K. S. Yang, and K. W. Kim, "Relationships between nonacoustic factors and subjective reactions to floor impact noise in apartment buildings," *Journal of the Acoustical Society of America*, vol. 139, pp. 1158–1167, 2016.

[4] "Floor Noise Management Center. Monthly Report (March 2018)," in `http://www.noiseinfo.or.kr/about/data_view.jspboardNo=199&keyfield=whole&keyword=&pg=2` 2018.

[5] H. Choi, H. Yang, S. Lee, and W. Seong, "Classification of Inter-Floor Noise Type/Position Via Convolutional Neural Network-Based Supervised Learning," *Applied Sciences*, vol. 9, no. 18, pp. 3735, 2019.

[6] A. J. Eronen, V. T. Peltonen, J. T. Tuomi, A. P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, "Audio-based context recognition," *IEEE Trans-*

actions on Audio, Speech, and Language Processing, vol. 14, no. 1, pp. 321–329, 2006.

[7] A. Rakotomamonjy and G. Gasso, "Histogram of gradients of time–frequency representations for audio scene classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 142–153, 2015.

[8] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 151–155, 2015.

[9] V. Bisot, R. Serizel, S. Essid, and G. Richard, "Acoustic scene classification with matrix factorization for unsupervised feature learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6445–6449, 2016.

[10] J. F. Gemmeke, L. Vuegen, P. Karsmakers, B. Vanrumste, "An exemplar-based nmf approach to audio event detection," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 1–4, 2013.

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

[12] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *IEEE 25th International Workshop on Machine Learning for Signal Processing*, pp. 1–6, 2015.

[13] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks," in *International Conference on Acoustics, Speech and Signal Processing*, pp. 559–563, 2015.

[14] 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.

[15] S. Hershey, S. Chaudhuri, D. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, “CNN architectures for large-scale audio classification,” in *International Conference on Acoustics, Speech and Signal Processing*, pp. 131-135, 2017.

[16] K. J. Piczak, “ESC: Dataset for environmental sound classification,” in *Proceedings of the ACM International Conference on Multimedia*, pp. 1015-1018, 2015.

[17] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, “Fma: A dataset for music analysis,” *arXiv preprint arXiv:1612.01840*, 2016.

[18] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, “The million song dataset,” *arXiv preprint arXiv:1612.01840*, 2011.

[19] S. Lee, H. Yang, H. Choi, and W. Seong, “Zero-Shot Single-Microphone Sound Classification and Localization in a Building via the Synthesis of Unseen Features,” *IEEE Transactions on Multimedia*, 2021.

[20] R. Vergin, and D. O’Shaughnessy, “Pre-emphasis and speech recognition,” in *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, pp. 1062-1065, vol. 2, IEEE, 1995.

[21] S. Ioffe, and C. Szegedy “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International Conference on Machine Learning*, pp. 448-456, vol. 37, 2015.

[22] D. A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (elus),” *arXiv preprint arXiv:1511.07289*, 2015.

Chapter 4

Zero-Shot Learning Approach for Identifying Type and Position of Sound Source

4.1 Introduction

4.1.1 Motivation

From the previous chapter, the annoying sounds were successfully classified based on their type and position under the supervised learning framework where we predict not only predict the type but also the position of the sound sources. In practice, the types are readily classified under the supervised learning frameworks with one-hot encoded labels. However, in practice, the positions are not compatible with those encoding schemes, since the sound source is positioned in a continuous space, which makes the number of classes for training uncountable. Further, significant effort is required to collect data to add new categories. Some places within a building are limited from collecting sufficient amounts of data, such as restricted areas. Therefore, the main focus of this chapter is to understand whether the positions of the sound sources can be properly predicted even for the locations that were not seen during training time. For this purpose, we introduce the zero-shot learning (ZSL) framework [1] which is an efficient learning framework to transfer the knowl-

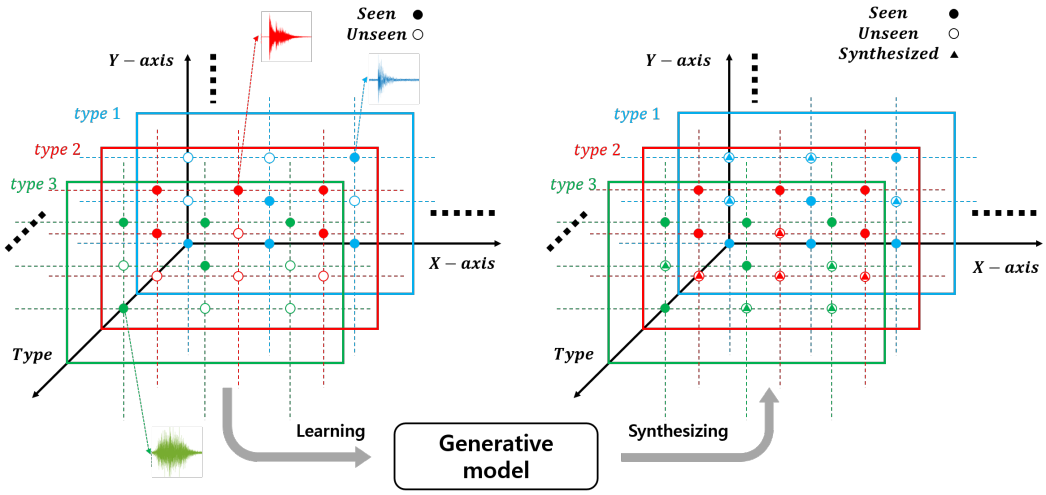


Figure 4.1: Audio examples are assumed to be randomly sampled from latent distributions that are conditioned on their class-embeddings [19]. Class-embeddings can be projected onto attribute coordinates, which indicate the types and positions of sound sources. For the ZSL setting, the classes are divided into seen (•) and unseen (○) classes. Learning a generative model conditioned on class-embeddings, unseen features can be synthesized from the attributes of unseen classes (▲).

edge from seen to unseen classes, and formulate the framework for our problem.

4.1.2 Related works

Zero-shot learning framework is inspired by the human ability to perceiver new semantics or concepts from what one has experienced and learned previously. In existing ZSL literatures, the semantics of classes are projected into semantic-embeddings or class-embeddings represented by human-annotated attributes [2, 3, 4], text descriptions [5], and word-embeddings [6, 7]. These class-embeddings are used as auxiliary information to transform information from the seen to unseen classes under the ZSL setting. Therefore, existing ZSL models generally focus on learning the mapping function between the feature representations of data and the corresponding

class-embeddings [2, 3, 4, 5, 6, 7, 8]. However, although the ZSL setting assumes that the test examples are restricted to unseen classes, considering the real world, test examples could arise from all classes, including seen and unseen. Hence, [9] established generalized zero-shot learning (GZSL) where the test examples can arise from, and be classified as, either seen or unseen classes. For such an extended problem, existing methods have been identified as inefficient [1, 9, 10], and efficient feature-generating approaches that have recently drawn attention [10, 11, 12, 13, 14, 15] with promising generative models, such as variational autoencoders (VAEs) [16], and generative adversarial networks (GANs) [17].

4.1.3 Contributions of this chapter

In this chapter, we propose a ZSL approach for identifying the types and positions of sound. As shown in Figure 4.1, we assume that audio examples are randomly generated from latent distributions conditioned on their corresponding class-embeddings, which can be projected onto the attribute coordinates. Class-invariant mapping between the attributes and audio examples should be learned from the seen examples. As the target classes can be either seen or unseen, we should verify the model's performance under the ZSL and GZSL settings. Thus, we focus on feature generation using GANs conditioned on class-embeddings. The class-embeddings are learned from the attributes in a higher-dimensional space along with the generative model. As the existing work [9] evaluated the performance under the ZSL and GZSL settings, our new indoor noise dataset, SNU-B36-EX, available at https://github.com/7tl7qns7ch/SNU-B36-EX, is randomly split according to their labels for the seen and unseen classes. Finally, we thoroughly evaluated the models on SNU-B36-EX with standard ZSL/GZSL settings.

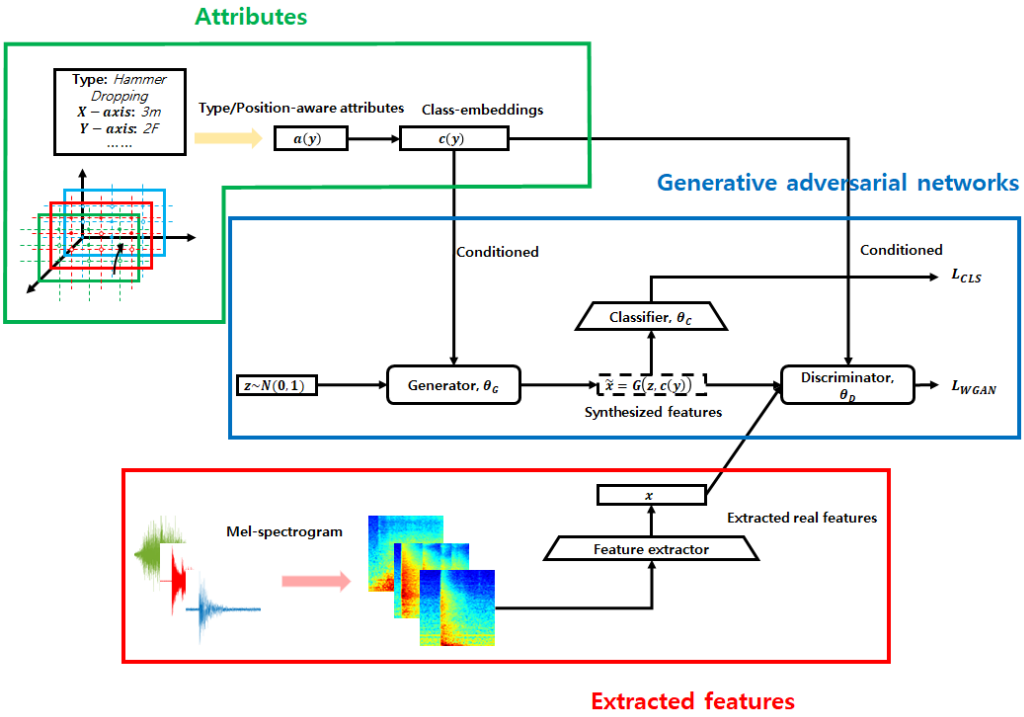


Figure 4.2: Overview of training of our generative model. The red box represents feature extraction, the yellow box represents 'type/position-aware attributes' and their projection on deep class-embedding at higher dimensions.

4.2 Approach

4.2.1 Problem formulation

Our given training set $S = \{(x_i^s, y_i^s, a(y_i^s)) | x_i^s \in \mathcal{X}^s, y_i^s \in \mathcal{Y}^s, a(y_i^s) \in \mathcal{A}^s\}_{i=1}^{N^s}$ where $x_i^s \in \mathcal{X}^s$ denotes an audio feature representation from seen classes; $y_i^s \in \mathcal{Y}^s$ denotes the corresponding label that is one of the seen classes $\mathcal{Y}^s$; and $a(y_i^s) \in \mathcal{A}^s$ denotes the corresponding attribute that is one of the seen attributes $\mathcal{A}^s = \{a(y) | y \in \mathcal{Y}^s\}$. Attribute $a(y)$ explains the semantic information of y in vector form. Using this information, we can model the relationship between all classes, and transfer the knowledge from the seen to unseen classes. For either ZSL or ZSSL, we are given classes that are assumed to be the unseen potential target classes. We are given sets of unseen classes $\mathcal{Y}^u$ and the corresponding unseen attributes $\mathcal{A}^u = \{a(y) | y \in \mathcal{Y}^u\}$. Here, the unseen classes are disjoint from the seen classes, that is, $\mathcal{Y}^s \cap \mathcal{Y}^u = \emptyset$. Unlike a set of seen data S, we cannot access features from unseen classes, i.e., $x_i^u \in \mathcal{X}^u$ is not available for the training set, but is available for the test set. Thus, we have class-level information on the unseen classes during the training phase.

During the training phase, the following empirical risk

$$E_{(x,y)\sim P(\mathcal{X}^s, \mathcal{Y}^s)} \mathcal{L}(f(x; W), y), \tag{4.1}$$

was minimized by training the training set S, where $\mathcal{L}$, and f are the loss and mapping functions parameterized by W, respectively. During the training phase, data from the seen classes can be used. We expect that the parameters from the aforementioned training process further minimize the following risks. The risks are defined through the trainset domain, for the ZSL setting as,

$$E_{(x,y)\sim P(\mathcal{X}^u, \mathcal{Y}^u)} \mathcal{L}(f(x; W), y), \tag{4.2}$$

when the test set consists of data from only unseen classes, and for the GZSL setting,

$$E_{(x,y)\sim P(\mathcal{X}^{s+u},\mathcal{Y}^{s+u})}\mathcal{L}(f(x;W),y), \tag{4.3}$$

when the test set consist of data from the seen and unseen classes. Here, $\mathcal{X}^{s+u} = \mathcal{X}^s \cup \mathcal{X}^u, \mathcal{Y}^{s+u} = \mathcal{Y}^s \cup \mathcal{Y}^u$.

Our goal is to model class-invariant mapping between classes and features using these attributes. We assume that the data are created by the class-invariant transformation of the corresponding attribute, and our target is to learn the transformation using a generative model. After training the generative model, it can synthesize the unseen features, and they can be utilized for immunization of our classifier before evaluating the test set. Figure 4.2 presents an overview of the training of our generative model.

4.2.2 Type/position-aware attributes

This section describes the annotations for the attributes $a(y)$, called "type/position-aware attributes". These attributes express the types and positions of the sound source in vector form. We assume that the distributions of the type and position are independent of each other. Thus, we can formulate a vector of attributes by splitting the components of the type and position of the sound source as follows:

$$a(y) = [t(y), p(y)] \in \mathbb{R}^{n_t+n_p}, \tag{4.4}$$

where $t(y) \in \mathbb{R}^{n_t}$ and $p(y) \in \mathbb{R}^{n_p}$ indicate the attributes of the source type and position between the source and receiver with dimensions n_t and n_p, respectively.

Throughout this study, because the source types are assumed to be from predefined types of sounds, the attributes of the source type are encoded by one-hot

vectors. Therefore, we define the attribute of the ith source type as

$$t(y) = \mathbb{1}_{n_t}(i) \in \mathbb{R}^{n_t}, \tag{4.5}$$

which denotes an n_t dimensional one-hot vector with only 1 for the ith element, and 0 for the others.

For the position of the sound source, we assume that the relative position between the source and receiver, $p(y)$, is present in k dimensional Euclidean space, as shown in Figure 4.1. Note that Figure 4.1 represents the case where $k = 2$. We can split $p(y)$ into k components, which indicate the corresponding spatial components.

$$\begin{gathered} p(y) = [p_1(y), p_2(y), ..., p_k(y)] \in \mathbb{R}^{n_p}, \\ p_m(y) \in \mathbb{R}^{n_{p_m}},\ m \in \{1, 2, ..., k\}, \qquad \sum_{m=1}^{k} n_{p_m} = n_p, \end{gathered} \tag{4.6}$$

where $p_1(y), p_2(y), ..., p_k(y)$ indicates the positional attribute for each spatial component with dimensions of $n_{p_1}, n_{p_2}, ..., n_{p_k}$, the sum of which is n_p. Furthermore, we describe two different annotations for positional attributes.

Multi-hot annotation

First, we can define the positional attribute for the mth spatial components as one-hot vector,

$$p_m(y) = \mathbb{1}_{n_{p_m}}(j_m),\ m \in \{1, 2, ..., k\}, \tag{4.7}$$

which denotes an n_{p_m} dimensional one-hot vector with 1 for the j_m^{th} element and 0 for the others; and j_m denotes one of the possible n_{p_m} positions in the mth spatial component. The annotation implies that the attributes make every position in each

spatial component orthogonal to each other. Thus, the whole attribute in this case is

$$a(y) = [\mathbb{1}_{n_t}(i), \mathbb{1}_{n_{p_1}}(j_1), \mathbb{1}_{n_{p_2}}(j_2), ..., \mathbb{1}_{n_{p_k}}(j_k)], \tag{4.8}$$

which is called a 'multi-hot' annotation. Annotation is considered as one of the multi-label schemes [18].

Linear annotation

We can provide a linear relationship between each positional attribute as,

$$p_m(y) = j_m = r_m^s - r_m^r,\ m \in \{1, 2, ..., k\}, \tag{4.9}$$

where r_m^s, and r_m^r denote the positions of the source and receiver projected on the mth spatial component, respectively. Unlike multi-hot annotation, it can provide the relative position between the source and receiver as a real number. Thus, the total dimension of the positional attribute n_p is equal to k. This annotation provides a spatially linear relationship between the positional attributes that are in the same spatial component. Thus, the whole attribute in this case is

$$a(y) = [\mathbb{1}_{n_t}(i), j_1, j_2, ..., j_k], \tag{4.10}$$

which is called a 'linear' annotation.

4.2.3 Zero-shot learning procedure

Deep class-embedding

The aforementioned attributes can be used as class-embeddings; however, class-embeddings can be projected on a more complex and higher dimension. Thus, deeper

class-embedding should be included in the end-to-end learning process using learnable networks from the attributes as input.

First, attributes are used as the following class-embedding [10]:

$$c(y) = a(y), \tag{4.11}$$

which is the baseline for further implementation. In this setting, the model is expected to learn the structure of class-embedding directly from the attributes. Second, we add an adaptive layer, A, parametrized by θ_A, which yields,

$$c(y) = A(a(y); \theta_A), \tag{4.12}$$

which is called an 'adaptive' layer. In this setting, the model is expected to learn the structure of the class-embedding in a higher-dimensional space from the attributes. Third, we replace the adaptive layer with separate parallel layers to learn each sub-concept, $A_0, A_1, ..., A_k$, parametrized by $\theta_{A_0}, \theta_{A_1}, ..., \theta_{A_k}$ for each type and positional component, respectively. The outputs are then concatenated as,

$$c(y) = [A_0(t(y); \theta_{A_0}), A_1(p_1(y); \theta_{A_1}), ..., A_k(p_k(y); \theta_{A_k})], \tag{4.13}$$

which is called a 'separative' layer. In this setting, the class-embedding is also learned in a high-dimensional space from the attributes, but each corresponding part of the class-embeddings is learned independently, and is then concatenated.

Feature generative model

Originally, a GAN consists of a generator G and a discriminator D, which are alternately trained as a minimax two-player game [17]. Following [10], the class-

invariant structure of the seen classes $c(y_s)$ can be learned through GANs conditioned with class-embeddings, as shown in Figure 4.2. The generative model can generate features of unseen classes through learned GANs conditioned on given unseen class-embeddings $c(y_u)$.

The model consists of a conditional generator $G : \mathcal{Z} \times \mathcal{C} \rightarrow \mathcal{X}$ parameterized by θ_G, and a conditional discriminator $D : \mathcal{X} \times \mathcal{C} \rightarrow [0, 1]$ parameterized by θ_D. Here, the generator G takes a random Gaussian noise vector $z \in \mathcal{Z} \sim \mathcal{N}(0, 1)$ and class-embedding $c(y) \in \mathcal{C}$, and yields fake features $\tilde{x}$ corresponding to class y. A feature representation x is then verified as real or fake by D conditioned on $c(y) \in \mathcal{C}$, yielding a value ranging from 0 to 1. The parameters of the generator and discriminator of GANs, that is, θ_G and θ_D, are alternately trained by minimizing the Wasserstein distance [19]. To train the model using the Wasserstein distance, the Lipschitz constraint of the solution should hold. Thus, we add the gradient penalty term to enforce the Lipschitz constraint, called WGAN-GP, which is a well-known method for stably training GANs [20]. The loss of the WGAN-GP is as follows:

$$\begin{aligned} \mathcal{L}_{WGAN}(\theta_G, \theta_D) = E[D(x, c(y_s); \theta_D)] - E[D(\tilde{x}, c(y_s); \theta_D)] \\ -\lambda E[(\|\nabla_{\hat{x}} D(\hat{x}, c(y_s); \theta_D)\|_2 - 1)^2], \end{aligned} \tag{4.14}$$

where $\tilde{x} = G(z, c(y_s); \theta_G)$ denotes fake features corresponding to the seen class-embeddings $c(y_s)$, $\hat{x} = \alpha x + (1 - \alpha)\tilde{x}$ with $\alpha \sim U(0, 1)$ denotes features sampled uniformly along straight lines between the real and fake feature distribution, and λ is the gradient penalty coefficient. Further, a classification loss is used to guarantee that the generated feature can be classified into the right class by a classifier parametrized by θ_C. The classification loss is expressed as follows:

$$L_{CLS}(\theta_G) = -E_{\tilde{x} \sim p_{\tilde{x}}}[\log P(y_s|\tilde{x}; \theta_C^*)], \tag{4.15}$$

where θ_C^* is pretrained by minimizing the classification loss using features from the

seen classes as supervised. Thus, the learning parameters of GAN, θ_G and θ_D, are estimated by optimizing the entire loss as a minimax game:

$$\theta_G^*, \theta_D^* = \arg\min_{\theta_G} \max_{\theta_D} L_{WGAN}(\theta_G, \theta_D) + \beta L_{CLS}(\theta_G), \tag{4.16}$$

where θ_G^* and θ_D^* are the optimal parameters for the generator and discriminator of the model trained by the seen classes, respectively; and β is a hyperparameter of the classifier loss weight.

Feature synthesis and classification

After training the generative model, we synthesize the same number (n) of unseen features for each unseen class using their class-embeddings, $\tilde{U} = \{(\tilde{x}_i^u, y_i^u, a(y_i^u)) | \tilde{x}_i^u \in \mathcal{X}^{\tilde{u}}, y_i^u \in \mathcal{Y}^u, a(y_i^u) \in \mathcal{A}^u\}_{i=1}^{N^u}$ where $\tilde{x}_i^u = G(z, c(y_i^u); \theta_G^*)$ and $\mathcal{X}^{\tilde{u}}$ denotes the synthesized feature distribution from unseen classes. By training the new training set, the following modified empirical risk can be minimized for the ZSL setting:

$$E_{(x,y)\sim P(\mathcal{X}^{\tilde{u}}, \mathcal{Y}^u)} \mathcal{L}(f(x; W), y), \tag{4.17}$$

which denotes that the classifier f parametrized by W can be trained with the synthesized features from unseen classes. For the GZSL setting, the synthesized features can be combined with features from the seen classes as a new training set. The combined set is utilized to train the classifier, f, to minimize the empirical risk for the GZSL setting:

$$E_{(x,y)\sim P(\mathcal{X}^{s+\tilde{u}}, \mathcal{Y}^{s+u})} \mathcal{L}(f(x; W), y), \tag{4.18}$$

where a feature can be from either seen features x_s or synthesized features $\tilde{x}^u$ from

unseen classes, that is, $\mathcal{X}^{s+\tilde{u}}$, and the corresponding class is one of all the classes, including the seen and unseen classes $\mathcal{Y}^{s+u}$. Instead of optimizing Equation (4.1), we expect that optimizing this modified objective is more effective in addressing the distribution discrepancy between the training and test set. In this study, we consider a softmax classifier with cross-entropy loss as the classifier.

4.3 Experimental Settings

4.3.1 Dataset preparation

Annotation

We annotate the 'type/position-aware attribute' on the dataset. We set the type attribute $t(y) = \mathbb{1}_{n_t}$ with $n_t = 5$ and each element indicates dropping a medicine ball on the floor (MB) for footstep sounds, dropping a hammer on the floor (HD), hitting with a hammer on the floor (HH) for hammering sounds, dragging a chair on the floor (CD) for sounds of dragging furniture, and operating a vacuum cleaner (VC) for electrical appliances as described in chapter 3, respectively. We set the positional attribute with $k = 2$ axes with $p_1(y)$ and $p_2(y)$ indicating the range (x-axis in Figure 3.3) and floor (y-axis in Figure 3.3), respectively. For example, when the sound source is ith type generated at j_1 meter on the j_2th floor, the 'multi-hot' annotation is $a(y) = [\mathbb{1}_5(i), \mathbb{1}_{13}(j_1), \mathbb{1}_3(j_2),] \in \mathbb{R}^{21}$ and 'linear' annotation is $a(y) = [\mathbb{1}_5(i), j_1, j_2 - 2] \in \mathbb{R}^7$.

Comparison with other ZSL datasets in the audio domain

As introduced in chapter 2, there are certain datasets for ZSL in the audio domain, including ESC-50 [22], FMA [23], and MSD [24]. We would like to compare our dataset with certain properties of ZSL settings. First, ESC-50 contains 2000 audio recordings of 50 classes, and Word2Vec is a 300-dimensional vector. The researchers of [25] split the dataset into 40 seen and 10 unseen classes. FMA contains 19,466 audio recordings of 157 classes and has 40-dimensional instrument vectors as attributes. MSD contains 406,409 audio recordings and 1126 classes, and GloVe is a 300-dimensional vector. In [26], FMA was split into 125 seen and 32 unseen classes, and MSD was split into 900 seen and 226 unseen classes. Table 4.1 summarizes the statistics of our dataset and comparisons. The primary difference is that SNU-B36-EX is for ZSL and ZSSL, while the others are for the ZSL task.

Table 4.1: Comparison of SNU-B36-EX with other ZSL datasets in the audio domain

Dataset	SNU-B36-EX	ESC-50	FMA	MSD
Audio	8,450	2,000	19,466	406,409
Class	169	50	157	1,126
Audio per class	50	40	$\approx$ 124	$\approx$ 361
Attribute	7 or 21	300	40	300
Task	ZSL, ZSSL	ZSL	ZSL	ZSL

Data split

For the standard ZSL and GZSL tasks, we split our 169 classes into 135 seen and 34 unseen classes with a data split scheme similar to [9], and the ratio between them was approximately $4:1$. The seen/unseen were randomly split by their labels, and 80% of the examples from each seen class are used as the training set. The remaining 20% of the examples from each seen class and all examples from unseen classes were used as the test set. Therefore, the number of data-points from the training and test sets were 5400 and 3050, respectively.

4.3.2 Evaluation metrics

We follow the evaluation metric proposed in [9] for our ZSL and GZSL settings. Under the ZSL setting, the classifier can predict audio examples of the test set to one of the unseen classes. Thus, the averaged per-class top-1 accuracy is computed only for unseen classes, denoted by $u \to u$. In the GZSL setting, the averaged per-class top-1 accuracy is computed for the seen classes, $s \to s+u$, and unseen classes, $u \to s+u$. Under this setting, the classifier can predict audio examples of the test set to any class, which is more challenging than for the ZSL setting. We also calculated the harmonic mean of $s \to s+u$ and $u \to u$, which is denoted by h.

4.3.3 Implementation details

We have two options for annotations: 'multi-hot' or 'linear', and two options for the deep class-embedding layer, 'adaptive' or 'separative', and the case where the attribute vector is used as the class-embeddings. For all the tasks, the architecture of the generator and discriminator was a single hidden layer with 1024 hidden units throughout this study. A single hidden layer with 512 hidden units consists of an 'adaptive' layer and 3 hidden layers with 170 hidden units as 'separative' layers. All the activation functions for GANs are ELU [27]. The noise vector z is sampled from a unit Gaussian distribution with 20 elements. We consider $\lambda = 10$ for the gradient penalty coefficient and $\beta = 0.5$ for the weight loss of the classifier. For every iteration of the generator, we consider five updates for the discriminator. All hyper-parameters were selected by trial and error, and early stopping was used to obtain the best performance. After training the GANs, to construct $\tilde{U}$, we synthesized the same number (n) of unseen features for each unseen class. We increased the number of n from 10 to 1000 during the experiments. To evaluate the classification performance of classifier W on the test set, the following statement should be clarified: Under the supervised learning setting (SUP), W is trained only on S and tested on the seen class examples from the test set. Under the ZSL setting, W is trained only on $\tilde{U}$ and tested on unseen class examples from the test set. Under the GZSL setting, W is trained on the combined set $S + \tilde{U}$ and tested on the entire test set.

4.3.4 Comparison of zero-shot learning methods

As mentioned in [1, 9, 10], traditional ZSL models, which usually learn the compatibility between different modalities, are significantly degraded under the GZSL setting. Thus, we compare our feature-generating methods with several non-generative methods under the ZSL/GZSL settings to verify the effectiveness of our models. ALE [3], ESZSL [4], SAE [8], and CMT [6] were selected for comparison.

4.4 Experimental Results

Table 4.2: Comparison of classification performance (%) for the feature representations under zero-shot learning.

Feature extractor		VGGish[28]	1D-CNN	VGGish [28] + 1D-CNN
Pre-trained on trainset		No	Yes	No+Yes
Dimension		128	128	256
ZSL	$u \rightarrow u$	50.23	51.18	**62.94**
GZSL	$u \rightarrow s+u$	37.00	40.26	**56.76**
	$s \rightarrow s+u$	43.33	58.15	**67.26**
	h	39.91	47.56	**61.57**

4.4.1 Audio feature representations

Although the SNU-B36-EX dataset was originally collected for zero-shot related tasks, we compared the representations under the ZSL/GZSL and supervised settings. In this case, 'linear' annotation for the attributes and 'adaptive' layer for the class-embedding were used to train the GANs. Table 4.2 compares the classification performance of the feature representations under the SUP/ZSL/GZSL settings. Using both feature representations was the best choice in all settings.

Comparing the individual feature representations, as the 1D-CNN is trained on the training set while the VGGish is not, the classification performance on the 1D-CNN features is 92.96%, which is better than that of 74.15% of the VGGish features under SUP. Despite the significant gap between the SUP performance results on the VGGish features and 1D-CNN features, the ZSL performance results for the two features (50.23% and 51.18%) are relatively close to each other. VGGish is pre-

trained with a much larger dataset, which ensures generalizability, whereas the 1D-CNN is more biased toward the training set. Under the GZSL setting, although the classification performance on both feature representations significantly dropped, the case of 1D-CNN, which has been accessed for seen classes, yields better results than VGGish. Considering that $\tilde{U}$ is constructed by the GANs trained with features from these extractors, these two extractors could be complementary to each other to extract meaningful features from our audio dataset.

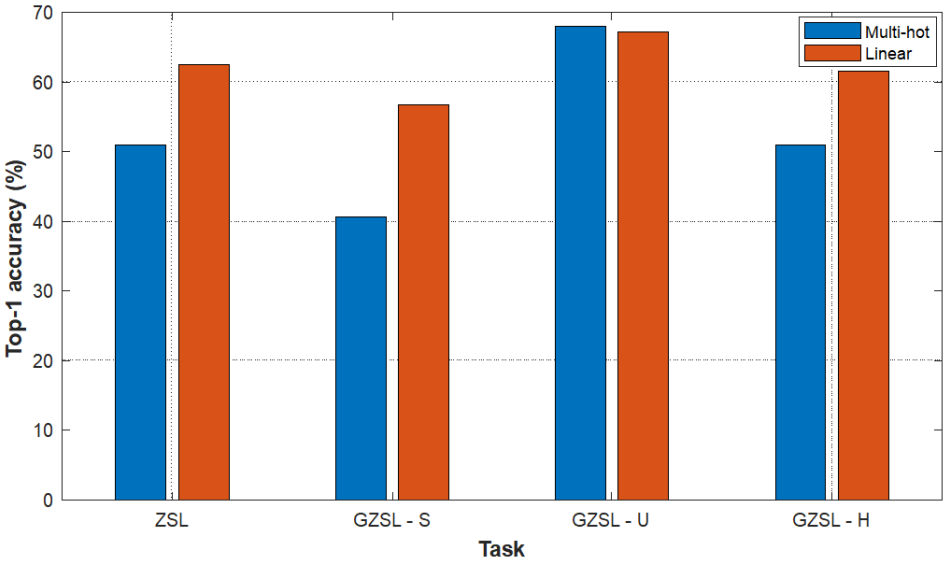


Figure 4.3: Comparison between two-annotation results (%) with 'adaptive' class-embedding under different tasks under the ZSL and GZSL settings.

4.4.2 Standard zero-shot/generalized zero-shot learning tasks

The results of the standard ZSL/GZSL tasks for our feature-generating models and other traditional ZSL models using two annotations are summarized in Table 4.3. For traditional methods, the performance of averaged top-1 accuracy $u \to u$, ranging from 14.47% to 45.12% under the ZSL setting and those on the harmonic mean, h, ranges from 3.94% to 25.19% under the GZSL setting. With our feature

Table 4.3: Performances (%) under the standard ZSL/GZSL settings.

Attribute	Model	ZSL	GZSL		
		$u \to u$	$u \to s+u$	$s \to s+u$	h
Multi-hot	ALE [3]	45.12	15.47	67.70	25.19
	ESZSL [4]	34.06	6.51	7.01	6.75
	SAE [8]	26.00	2.76	13.56	4.59
	CMT [6]	38.24	8.65	64.59	15.25
	FCLSGAN [10]	60.47	51.94	65.19	57.81
	+Adaptive	50.94	40.76	68.00	50.97
	+Separative	61.41	47.24	67.56	55.60
Linear	ALE [3]	27.06	3.59	7.78	4.91
	ESZSL [4]	34.12	5.92	7.21	6.50
	SAE [8]	14.47	3.88	4.00	3.94
	CMT [6]	30.59	11.41	31.33	16.73
	FCLSGAN [10]	56.47	48.12	63.85	54.88
	+Adaptive	**62.53**	**56.76**	67.26	**61.57**
	+Separative	61.12	52.47	**68.15**	59.29

generative model, the performance of the averaged top-1 accuracy $u \to u$, ranging from 50.94% to 62.53% under the ZSL setting, and that of the harmonic mean h ranges from 50.97% to 61.57% under the GZSL setting. The feature generative model with 'linear' annotation and 'adaptive' class-embedding results in the best performance on averaged top-1 accuracy of $u \to u$, under the ZSL setting and the harmonic mean h, under the GZSL setting.

Comparing the two annotations for our feature-generating methods, 'linear' annotation results in relatively better performance when compared with the 'multi-hot' annotation when the deep class-embeddings are modeled by the 'adaptive' layer to classify unseen classes under both ZSL and GZSL settings. Hence, 'linear' annotation would be more beneficial for transferring the knowledge on the attribute coordinates in these tasks. With 'multi-hot' annotation, each subcomponent of the attribute is designed as a one-hot vector, which is less informative for understanding the relations of the classes at latent space. The traditional methods tend to show better performance when the attributes are given by 'multi-hot' annotations and significantly lose their generalization ability under the GZSL settings for both annotations when compared with our feature-generating methods. In particular, in the case of ALE [3] and CMT [6] with 'multi-hot' annotations, the performance on seen classes $s \to s+u$ (67.70% and 64.59%) was comparable to that of our feature-generating methods. However, the performance of the unseen classes $u \to s+u$ of the cases is significantly degraded compared to that of our feature-generating methods. Comparing the three deep class-embeddings in this task, while the results of the deep class-embeddings vary depending on the annotations, the 'adaptive' layer is the best choice to comprehend the information of the 'linear' annotated attribute in deep embedding space.

4.4.3 Effects of the number of unseen features per class

Under the GZSL settings, to appropriately combine the training set and synthesized unseen set $\tilde{U}$, we examine the performance sensitivity to the size of $\tilde{U}$.

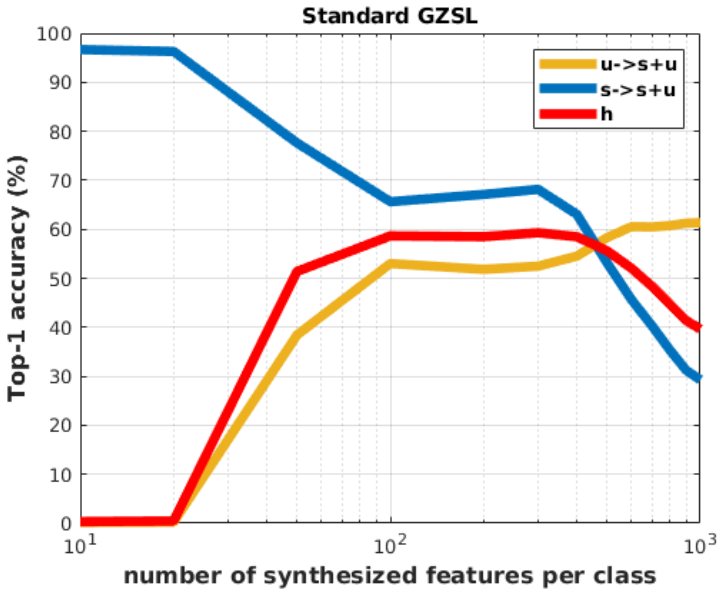


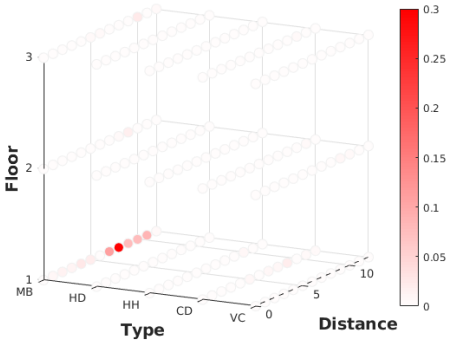
Figure 4.4: Effects of the number of unseen features per class under standard GZSL.

Figure 4.4 shows examples of the results of the averaged top-1 accuracies of the seen ($s \to s+u$) and unseen classes ($u \to s+u$), and their harmonic means h as a function of the number of synthesized unseen features per class n, varying from 10 to 1000 for each task. The x-axis of the plot is log-scaled. The harmonic mean h increases from $n = 10$ to $n = 100$ and becomes relatively flat until approximately $n = 400$ or 500 and decreases. As n increases, the averaged top-1 accuracy of the seen classes ($s \to s+u$) tends to decrease from approximately 96%, whereas that of the unseen classes ($u \to s+u$) tends to increase, starting from approximately 0%. When the number of synthesized unseen features is small, the classifier has the capability of seen features with similar performance under SUP, but not that of classifying the unseen features. By increasing the number of synthesized unseen features, the classifiers would gradually shift their classification ability, which is overfitted to the observed features, to classify unseen features. That is, as the size of $\tilde{U}$ increases, the classifiers gain the ability to classify unseen features and lose the ability to classify seen features for generalization to all classes. However, if the

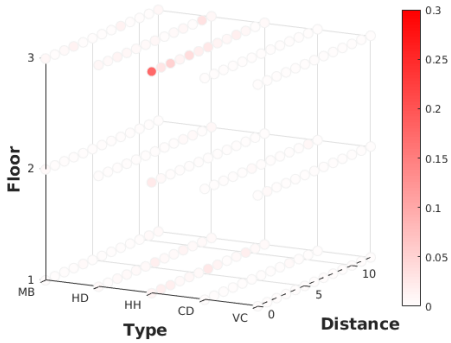
number of synthesized unseen features exceeds a certain value, the classifiers lose their classification ability on the seen classes, whereas the ability on unseen classes is saturated. Therefore, to classify both classes generally, the appropriate number of unseen features that should be synthesized is $n = 200 - 500$.

4.4.4 Visualization of the classifier's prediction

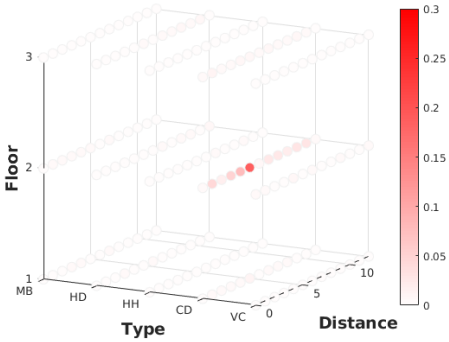
Figure 4.5 and 4.6 show examples of the averaged per-class softmax outputs of the classifier under the standard GZSL setting. The sounds of 'dropping a medicine ball at 8 m on the 1st floor', 'hitting with a hammer at 0m on the 3rd floor', 'dragging a chair at 5 m on the 2nd floor', and 'operating a VC at 12 m on the 2nd floor' are from seen classes, and 'dropping a medicine ball at 8 m on the 3rd floor', 'dropping a hammer at 9 m on the 2nd floor', 'hitting with a hammer at 1m on the 2nd floor', and 'dragging a chair at 9 m on the 1st floor' are from unseen classes, respectively. Thus, our method predicts values close to the ground truth values for the seen and unseen classes. However, the targets from the seen classes have sharp probabilities, whereas those from the unseen classes have relatively smooth probabilities. This means that the classifier can identify examples from seen classes with higher confidence than those from unseen classes. Probability values of non-target classes are similar to the target class.



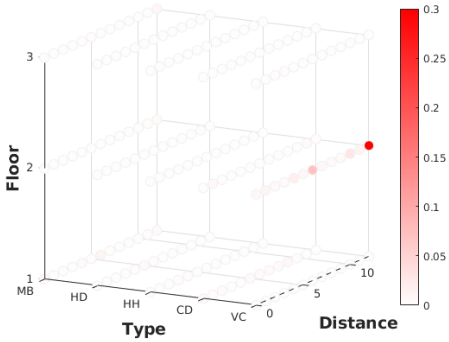
(a) MB8M1F - seen class



(b) HH0M3F - seen class

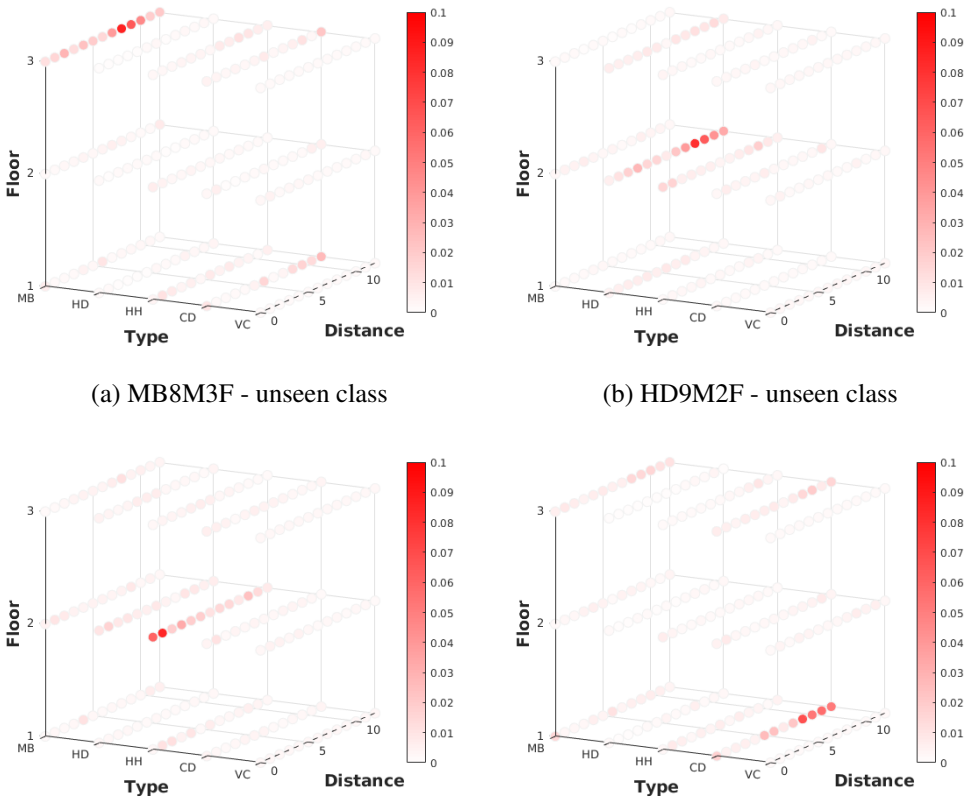


(c) CD5M2F - seen class



(d) VC12M2F - seen class

Figure 4.5: Seen examples of the averaged per-class softmax outputs of the zero-shot classifier.



(a) MB8M3F - unseen class

(b) HD9M2F - unseen class

(c) HH1M2F - unseen class

(d) CD9M1F - unseen class

Figure 4.6: Unseen examples of the averaged per-class softmax outputs of the zero-shot classifier.

4.5 Conclusion

We attempted to simultaneously estimate the type and position of an indoor sound source using a zero-shot learning framework. Vectorizing the concepts of classes with 'type/position-aware attributes' and capturing them with deep class-embeddings were proposed to encourage our generative models to learn a more reasonable class-invariant mapping from attributes to features. Thus, our generative models trained with seen examples can be used to synthesize unseen features from attributes of unseen classes that are the potential target positions that are not accessible during training. The synthesized unseen features were used to train classifiers that have the capability of classifying features from the seen and unseen classes. Comprehensive experiments, including the comparison of feature extractors, different encoding methods of the 'type/position-aware attributes' and class-embeddings, various configurations of seen/unseen data, and other zero-shot learning methods, are conducted using the new indoor noise dataset, SNU-B36-EX. The best performance is about 62.53 % under the ZSL setting and 61.57 % under the GZSL setting, when some parts of classes are not available during the training. In the later chapter, we would like to apply the zero-shot learning frameworks for thoroughly verifying the knowledge transferability from seen to unseen positions in several cases.

Bibliography

[1] Y. Fu, T. Xiang, Y.-G. Jiang, X. Xue, L. Sigal, and S. Gong, "Recent advances in zero-shot recognition: Toward data-efficient understanding of visual content," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 112–125, 2018.

[2] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 453–465, 2013.

[3] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1425–1438, 2015.

[4] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *International Conference on Machine Learning*, pp. 2152–2161, 2015.

[5] L. Zhang, T. Xiang, and S. Gong, "Learning a deep embedding model for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[6] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," in *Advances in Neural Information Processing Systems*, pp. 935–943, 2013.

[7] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, "Devise: A deep visual-semantic embedding model," in *Advances in Neural Information Processing Systems*, pp. 2121–2129, 2013.

[8] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3174–3183, 2017.

[9] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2251–2265, 2018.

[10] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5542–5551, 2018.

[11] V. Kumar Verma, G. Arora, A. Mishra, and P. Rai, "Generalized zeroshot learning via synthesized examples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4281–4289, 2018.

[12] E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata, "Generalized zero-and few-shot learning via aligned variational autoencoders," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8247–8255, 2019.

[13] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, and A. Elgammal, "A generative adversarial approach for zero-shot learning from noisy texts," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1004–1013, 2018.

[14] R. Felix, V. B. Kumar, I. Reid, and G. Carneiro, "Multi-modal cycleconsistent generalized zero-shot learning," in *Proceedings of the European Conference on Computer Vision*, pp. 21–37, 2018.

[15] Y. Xian, S. Sharma, B. Schiele, and Z. Akata, "f-vaegan-d2: A feature generating framework for any-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10275–10284, 2019.

[16] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

[18] M. L. Zhang and Z. H. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2013.

[19] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[20] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, pp. 5767–5777, 2017.

[21] S. Lee, H. Yang, H. Choi, and W. Seong, "Zero-Shot Single-Microphone Sound Classification and Localization in a Building via the Synthesis of Unseen Features," *IEEE Transactions on Multimedia*, 2021.

[22] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Proceedings of the ACM International Conference on Multimedia*, pp. 1015-1018, 2015.

[23] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "Fma: A dataset for music analysis," *arXiv preprint arXiv:1612.01840*, 2016.

[24] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," *arXiv preprint arXiv:1612.01840*, 2011.

[25] H. Xie and T. Virtanen, "Zero-shot audio classification based on class label embeddings," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, IEEE, pp. 264–267, 2019.

[26] J. Choi, J. Lee, J. Park, and J. Nam, "Zero-shot learning for audio-based music classification and tagging," *arXiv preprint arXiv:1907.02670*, 2019.

[27] D. A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[28] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, "Cnn architectures for large-scale audio classification," in *International Conference on Acoustics, Speech and Signal Processing*, IEEE, pp. 131–135, 2017.

Chapter 5

Knowledge Transferability Through Positions of Sound Source

5.1 Introduction

5.1.1 Motivation

From the previous chapters, the annoying sounds were successfully classified based on their type and position under both supervised and zero-shot learning frameworks where we predict not only predict the seen classes during the training but also the unseen classes during the training. Meanwhile, because the sounds are assumed to come from pre-defined types, the seen and unseen classes have similar type distribution. Therefore, the discrepancy between the seen and unseen classes is dominated by the position of the sounds. A simple random data split according to their classes could not completely separate the positions into seen and unseen classes. Thus, we should further validate the model with another seen/unseen data split scheme to determine the knowledge transferability of the methods from seen positions to unseen positions, called a zero-shot source localization (ZSSL) setting.

5.1.2 Related works

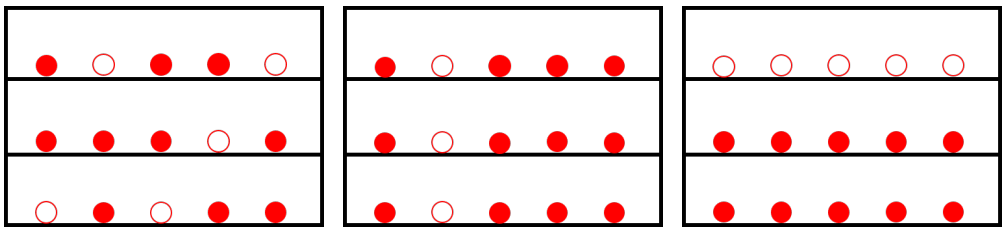
Sound source localization (SSL), especially for indoor sound or sound inside a building, has been widely studied in the literatures. Most existing works on SSL obtain signals of indoor sounds from multiple sensors or microphone arrays with known geometry to utilize time/phase delays [1, 2], reverberation [3, 4, 5], and energy-based information [6, 7]. Recently, there have been many attempts to solve SSL problem using learning-based approaches in which the neural networks are trained with non -informative noise sources [8], diverse sound events [9, 10], speech of speakers [11, 12, 13], or simulated data [14]. However, they are not compatible with our problem, in which audio signals are recorded using a single microphone. Additionally, in sound event localization and detection (SELD), a spatio-temporal characterization of the acoustic scene is obtained by combining the sound event detection (SED) and SSL [15]. In earlier works on SELD, two problems are separately treated with off-the-shelf module for detection, and classic array processing methods for localization [16, 17, 18]. In recent years, many attempts have been made to utilize neural networks as learning based approaches for the SELD problems. The joint probabilities for each type and position of the sound are predicted by a convolutional neural networks (CNN) or convolutional recurrent neural networks (CRNN) as a multi-label classification problem [19, 20, 21, 22].

5.1.3 Contributions of this chapter

Our problem is similar to SELD, in which we are interested in both the identification of the type and position of sounds. However, in terms of the output format, the types of sound are classified per signal, not SED, which aims to detect the frame-wise occurrence of the sound events. In addition, while the existing SELD uses multi-channel signals as the dataset, signals are given as a single channel in our problem. In this chapter, we propose extended tasks, called zero-shot source localization, to validate the knowledge transferability through positions of the sound

with the learned models. Under the ZSSL setting, the seen/unseen classes were randomly split according to their positions. The sole difference between ZSL and ZSSL is the data split scheme, as the ZSSL setting is expected to thoroughly validate the knowledge transferability along with the distributed positions. Further, the generalized zero-shot sound source localization (GZSSL) test set of ZSSL contains seen and unseen classes. In the ZSSL/GZSSL settings, three different cases are considered to be our testbed, which is described in Figure 5.1.

5.2 Approach



(a) Case 1 (b) Case 2 (c) Case 3

Figure 5.1: Schematic elevation view of three different cases for the ZSSL tasks. The classes are randomly split into seen (•) and unseen (○) classes according to (a) position, (b) range, and (c) floor.

As mentioned in the introduction of this chapter, since the only difference between ZSL and ZSSL is the data split scheme, the training procedures of architecture including GANs and classifiers are the same as in the previous chapter. Like the standard ZSL and GZSL tasks, the seen/unseen were randomly split by their labels, similar to [23]. Also, 80% of the examples from each seen class is used as the training set and the remaining 20% of the examples from each seen class and all examples from unseen classes were used as the test set.

For the ZSSL and GZSSL tasks, we consider three different cases as follows.

1. Case 1: All classes are randomly split by their position (Figure 5.1 (a)). There

are 39 possible positions in SNU-B36-EX, which are split into 31 seen classes and 8 unseen classes. Therefore, a total of 169 classes are divided into 135 seen and 34 unseen classes.

2. Case 2: All classes are randomly split by their range Figure 5.1 (b)). There are 13 possible range values, and we split them into 10 seen and 3 unseen classes. Therefore, a total of 169 classes are divided into 130 seen and 39 unseen classes.

3. Case 3: All classes are randomly split by their floor (Figure 5.1 (c)). Sound sources can be present on three possible floors. We choose the 3rd floor as unseen classes and the others as seen classes. Therefore, a total of 169 classes were divided into 117 seen and 52 unseen classes.

Case 1 is the verification of the knowledge transferability of the methods from seen positions to unseen positions when the seen positions are randomly distributed in the building. Cases 2 and 3 verify the knowledge transferability along the x-axis and y-axis, respectively.

5.3 Experiments

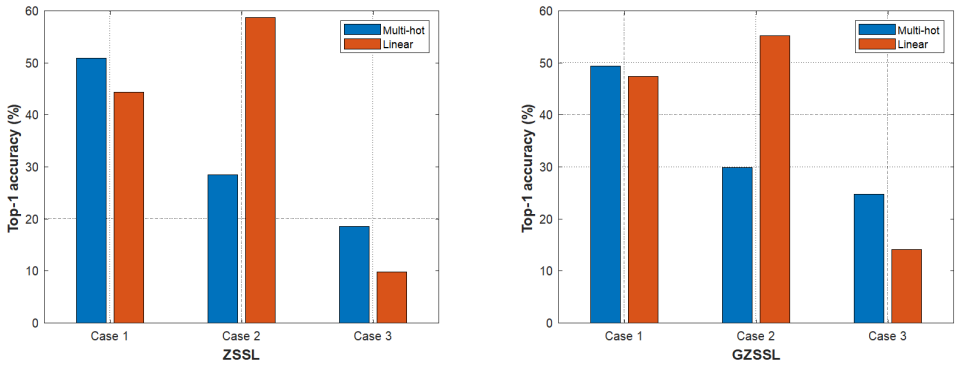
5.3.1 Experimental settings

We follow the evaluation metric proposed in [23] for our ZSSL and GZSSL settings. In this study, as the only difference between ZSL and ZSSL (and GZSL and GZSSL) is the data split scheme, we consider the same evaluation metric for the ZSSL and GZSSL tasks. Therefore, the averaged per-class top-1 accuracy is computed only for unseen classes under the ZSSL settings, denoted $u \to u$. In the GZSSL setting, the averaged per-class top-1 accuracy is computed for the seen classes, $s \to s+u$, and unseen classes, $u \to s+u$. Additionally, the harmonic mean of $s \to s+u$ and $u \to s+u$ is computed as h.

All the implementation details and hyperparameters are the same as ZSL/GZSL settings, the only difference is the feature extractors of each case are pretrained by the corresponding seen classes. Therefore, after training the corresponding generative models, to construct $\tilde{U}$, we synthesized the same number (n) of unseen features for each unseen classes. We increased the number of n from 10 to 1000 during the experiments. In order to evaluate the classification performance of classifiers on the respective test set, each classifiers is trained only on seen classes and tested on the seen class examples from test set. Under the ZSSL setting, each classifier is trained only on synthetic unseen data $\tilde{U}$ and tested on unseen class examples from the test set. Under the GZSSL setting, each classifiers is trained on the combined set $S + \tilde{U}$ and tested on the entire test set.

5.3.2 Experimental results

Table 5.1, 5.2, and 5.3 summarize the results under the ZSSL/GZSSL settings with three data split schemes for our models using two annotations as side information. The 'adaptive' layer for the deep class-embeddings yields the best results on the averaged top-1 accuracy of $u \to u$ and the harmonic mean h, when 'multi-hot'



(a) ZSSL settings

(b) GZSSL settings

Figure 5.2: Comparison between two-annotation results (%) with 'adaptive' class-embedding under different tasks with (a) ZSSL settings, and (b) GZSSL settings.

annotation is given under cases 1 and 3, and 'linear' annotation is given in case 2. Figure 5.2 compares the performance of annotations with an 'adaptive' layer for the deep class-embeddings under different tasks.

In case 1, where the seen/unseen classes are randomly split by their positions, the performance of averaged top-1 accuracy $u \to u$ ranges from 38.82% to 50.88% under the ZSSL setting, and for the harmonic mean, h, ranges from 34.91% to 49.33% under the GZSSL setting. Compared with standard ZSL/GZSL settings, the performance dropped for both annotations, even though the ratios of seen/unseen classes were the same. This is because the positional attributes of unseen classes in case 1 are totally blocked during training, while the positional attributes of unseen classes under standard ZSL/GZSL can be learned implicitly by another seen class with different types and at the same position. In particular, the performance of 'linear' annotation is more significantly degraded and worse than that of 'multi-hot' annotation. This means that it is beneficial to transfer the knowledge from the seen position to the unseen position with 'multi-hot' annotation in this case.

In case 2, where the seen/unseen classes are randomly split by their range, the

Table 5.1: ZSSL/GZSSL performances (%) under case 1 data split scheme.

Attribute	Model	ZSSL	GZSSL		
		$u \to u$	$u \to s+u$	$s \to s+u$	h
Multi-hot	FCLSGAN	39.47	33.24	70.44	45.16
	+Adaptive	**50.88**	38.47	68.74	**49.33**
	+Separative	43.94	35.18	68.07	46.38
Linear	FCLSGAN	38.82	31.76	38.74	34.91
	+Adaptive	44.41	37.47	64.44	47.39
	+Separative	45.18	37.12	61.04	46.16

performance of averaged top-1 accuracy $u \to u$ ranges from 28.51% to 58.67% under the ZSL setting, and for the harmonic mean, h, ranges from 29.89% to 55.24% under the GZSL setting. The performance with 'linear' annotation is better than that with 'multi-hot' annotation. Compared with standard ZSL/GZSL settings, the performance with 'multi-hot' is significantly degraded, while that with 'linear' drops slightly. Regarding the seen/unseen classes being split by their range in this case, 'linear' annotation is more appropriate than 'multi-hot' annotation to model the range components (x-axis in Figure 3.3) of the positional attributes for transferring knowledge from seen to unseen classes.

In case 3, where the seen/unseen classes are randomly split by their floor, the performance of the averaged top-1 accuracy $u \to u$ ranges from 9.77% to 18.62% under the ZSL setting, and those on harmonic mean, h, ranges from 13.28% to 24.78% under the GZSL setting. The performance with both annotations is significantly degraded compared with the standard ZSL/GZSL settings. A reason for this significant performance degradation is the proportion of unseen to whole classes split. In this case, the proportion of unseen classes is approximately $30.8\%(52/169)$, while

Table 5.2: ZSSL/GZSSL performances (%) under case 2 data split scheme.

Attribute	Model	ZSSL	GZSSL		
		$u \to u$	$u \to s+u$	$s \to s+u$	h
Multi-hot	FCLSGAN	35.18	28.62	56.46	37.98
	+Adaptive	28.51	22.05	46.38	29.89
	+Separative	31.85	23.64	51.08	32.32
Linear	FCLSGAN	57.33	47.08	65.54	54.79
	+Adaptive	**58.67**	48.77	63.69	**55.24**
	+Separative	58.05	44.67	69.85	54.49

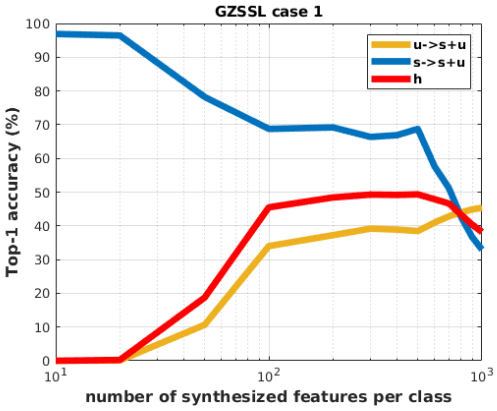
the others are approximately $20.1\%(34/169)$ and $23.1\%(39/169)$. Furthermore, the sounds on the 3rd floor, which are from unseen classes, have fairly different distributions from those on the 1st and 2nd floors, which are trained as seen classes. Despite the significant drop in both annotations, the performance with 'linear' annotation is more degraded compared with that having 'multi-hot' annotation. Hence, the floor components (y-axis in Figure 3.3) of the positional attribute are more likely to be independent of each other in the attribute space.

Under the GZSSL settings, to appropriately combine the training set and synthesized unseen set $\tilde{U}$, we examine the performance sensitivity to the size of $\tilde{U}$. Figure 5.3 shows examples of the results of the averaged top-1 accuracies of the seen ($s \to s+u$) and unseen classes ($u \to s+u$), and their harmonic means h as a function of the number of synthesized unseen features per class n, varying from 10 to 1000 for each task. The x-axis of the plot is log-scaled. The harmonic mean h increases from $n = 10$ to $n = 100$ and becomes relatively flat until approximately $n = 400$ or 500 and decreases, except for GZSSL case 3. As n increases, the averaged top-1 accuracy of the seen classes ($s \to s+u$) tends to decrease from approximately 96%,

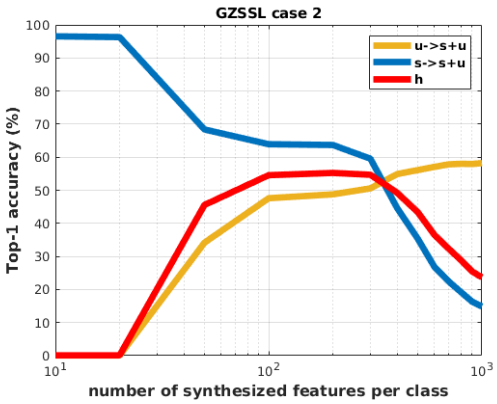
Table 5.3: ZSSL/GZSSL performances (%) under case 3 data split scheme.

Attribute	Model	ZSSL	GZSSL		
		$u \to u$	$u \to s+u$	$s \to s+u$	h
Multi-hot	FCLSGAN	16.35	15.15	61.88	24.35
	+Adaptive	**18.62**	16.23	52.39	**24.78**
	+Separative	16.23	15.31	58.89	24.30
Linear	FCLSGAN	13.42	11.69	38.80	17.97
	+Adaptive	9.77	8.81	35.98	14.15
	+Separative	9.77	7.88	42.05	13.28

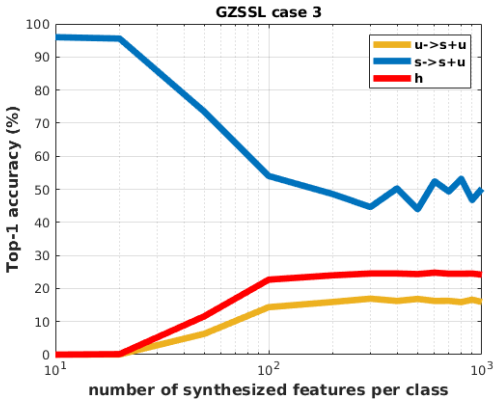
whereas that of the unseen classes ($u \to s + u$) tends to increase, starting from approximately 0%. When the number of synthesized unseen features is small, the classifier has the capability of seen features with similar performance under SUP, but not that of classifying the unseen features. By increasing the number of synthesized unseen features, the classifiers would gradually shift their classification ability, which is overfitted to the observed features, to classify unseen features. That is, as the size of $\tilde{U}$ increases, the classifiers gain the ability to classify unseen features and lose the ability to classify seen features for generalization to all classes. However, if the number of synthesized unseen features exceeds a certain value, the classifiers lose their classification ability on the seen classes, whereas the ability on unseen classes is saturated. Therefore, to classify both classes generally, the appropriate number of unseen features that should be synthesized is $n = 200 - 500$.



(a) GZSSL case 1



(b) GZSSL case 2



(c) GZSSL case 3

Figure 5.3: Effects of the number of unseen features per class under different tasks.

5.4 Conclusion

We attempted to validate the model with extended tasks when the seen/unseen classes are separated by their positions, which is called zero-shot source localization (ZSSL). We proposed three different data split cases under the ZSSL/GZSSL tasks. The results indicate that the knowledge transferability of the model is effective through the range direction (x-axis), but not effective through the floor direction (y-axis). The results can be interpreted that there can be a huge gap in the data from the different floors. The results could be more established by the further future works with more audio data from other floors.

Bibliography

[1] H. Atmoko, D. Tan, G. Tian, and B. Fazenda, "Accurate sound source localization in a reverberant environment using multiple acoustic sensors," *Measurement Science and Technology*, vol. 19, no. 2, 2008.

[2] X. Alameda-Pineda and R. Horaud, "A geometric approach to sound source localization from time-delay estimates," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 6, pp. 1082–1095, 2014.

[3] T. Gustafsson, B. D. Rao, and M. Trivedi, "Source localization in reverberant environments: Modeling and statistical analysis," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 791–803, 2003.

[4] F. Ribeiro, C. Zhang, D. A. Florencio, and D. E. Ba, "Using reverberation to improve range and elevation discrimination for small array sound source localization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1781–1792, 2010.

[5] I. An, D. Lee, J. Choi, D. Manocha, and S. Yoon, "Diffractionaware sound localization for a non-line-of-sight source," in *International Conference on Robotics and Automation*, IEEE, pp. 4061–4067, 2019.

[6] X. Sheng and Y.-H. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 53, no. 1, pp. 44–53, 2004.

[7] F. Deng, S. Guan, X. Yue, X. Gu, J. Chen, J. Lv, and J. Li, "Energybased sound source localization with low power consumption in wireless sensor networks," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 4894–4902, 2017.

[8] S. Chakrabarty and E. A. P. Habets, "Multi-speaker DOA estimation using deep convolutional networks trained with noise signals," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 8–21, 2019.

[9] T. N. T. Nguyen, W. S. Gan, R. Ranjan, and D. L. Jones, "Robust source counting and doa estimation using spatial pseudo-spectrum and convolutional neural network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2626–2637, 2020.

[10] J. Pak and J. W. Shin, "Sound localization based on phase difference enhancement using deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1335–1345, 2019.

[11] R. Takeda and K. Komatani, "Sound source localization based on deep neural networks with directional activate function exploiting phase information," in *International Conference on Acoustics, Speech and Signal Processing*, IEEE, pp. 405–409, 2016.

[12] R. Takeda, Y. Kudo, K. Takashima, Y. Kitamura, and K. Komatani, "Unsupervised adaptation of neural networks for discriminative sound source localization with eliminative constraint," in *International Conference on Acoustics, Speech and Signal Processing*, IEEE, pp. 3514–3518, 2018.

[13] L. Perotin, R. Serizel, E. Vincent, and A. Guerin, "Crnn-based multiple doa estimation using acoustic intensity features for ambisonics recordings," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 22–33, 2019.

[14] W. He, P. Motlicek, and J.-M. Odobez, "Adaptation of multiple sound source localization neural networks with weak supervision and domainadversarial train-

ing," in *International Conference on Acoustics, Speech and Signal Processing*, IEEE, pp. 770–774, 2019.

[15] A. Politis, A. Mesaros, S. Adavanne, T. Heittola, and T. Virtanen, "Overview and evaluation of sound event localization and detection in dcase 2019," 2020.

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

[17] T. Butko, F. G. Pla, C. Segura, C. Nadeu, and J. Hernando, "Two source acoustic event detection and localization: Online implementation in a smart-room," in *19th European Signal Processing Conference*, pp. 1317–1321, 2011.

[18] K. Lopatka, J. Kotus, and A. Czyzewski, "Detection, classification and localization of acoustic events in the presence of background noise for acoustic surveillance of hazardous situations," *Multimedia Tools and Applications*, vol. 75, pp. 10407–10439, 2015.

[19] T. Hirvonen, "Classification of spatial audio location and content using convolutional neural networks," vol. 2, 05 2015.

[20] K. Noh, C. Jeong-Hwan, J. Dongyeop, and C. Joon-Hyuk, "Threestage approach for sound event localization and detection," *Tech. report of Detection and Classification of Acoustic Scenes and Events 2019 (DCASE) Challange*, 2019.

[21] R. Varzandeh, K. Adiloglu, S. Doclo, and V. Hohmann, "Exploiting periodicity features for joint detection and doa estimation of speech sources using convolutional neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 566–570, 2020.

[22] F. Ronchini, D. Arteaga, and A. Perez-Lopez, “Sound event localization and detection based on crnn using rectangular filters and channel rotation data augmentation,” *arXiv preprint arXiv:2010.06422*, 2020.

[23] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, “Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2251–2265, 2018.

Chapter 6

Conclusion

Contributions of this dissertation are as follows:

- First, we propose a learning-based approach to the type and position of sounds and introduce our new dataset, SNU-B36-EX, collected in a real-world building using a single microphone. Then, the experiment shows that the datasets can be well classified with modern deep architectures, such as convolutional neural networks.
- Second, we raise potential issues on the generalization ability of the existing learning-based methods for sound classification and localization when the part of data is limited. Then, we attempt to improve the generalization ability of the model with efficient learning frameworks, called zero-shot learning, pursuing data efficiency which can make the model robust on unseen data during the training.
- Third, we observe that the types of sounds are assumed to be pre-defined types, the most discrepancies between the seen and unseen classes are caused by the position of the sounds. Therefore, we attempted to validate the model with extended experiments where the seen and unseen classes are separated by their positions, which is called zero-shot source localization.

We apply the supervised learning framework with two convolutional modules, such as task-specific and task-general, to identify the type and position of the sound source as a joint classification problem. The experimental results show that (1) our datasets are successfully classified by the model, which further can be used as feature extractors for transfer-learning, and (2) the task-specific and task-general feature extractors are complementary to each other.

We apply the zero-shot learning framework to learn shared representations between audio signals and the corresponding classes as evaluating the methods on the real-world datasets for source localization and classification problems. The experimental results show that (1) feature representation of new data can be synthesized from previously accessible data with the attributes of the system inputs and promising generative model, and (2) the synthesized unseen features contain sufficient information to classify the seen and unseen classes of the test set, which can be interpreted as some information are transferable from seen to unseen classes through the proposed methods.

We proposed three different data split settings for the zero-shot source localization tasks. The experimental results indicate that (1) knowledge transferability of the model is effective through the range direction (x-axis), but not effective through the floor direction (y-axis), which can be interpreted that there can be a huge gap in the data from the different floors, and (2) therefore, the proposed methods are robust to learn totally new data from a novel combination of positions and type, which makes it possible to treat source localization and classification problem in a data-driven way.

We expect that these procedures could be extended to general SELD problems where the potential target position is not restricted to grid-like points, as in our problem. Furthermore, the system would have the versatility to robustly classify unidentified sounds for other buildings by extending the dataset by collecting the sounds from more than one building.

The superficial meaning of zero-shot learning frameworks is merely to pursue sampling efficiency and reduce the time complexity of training the neural networks, but the frameworks truly seek to the way of construct the global pattern beyond a single instance of the system. From the perspective of establishing the patterns of the sound propagation along with the complex building structure, these directions of studies should be encouraged for the sound classification and localization as pure data-driven methods rather than naive supervised learning which inconsiderately requires massive data and expensive training procedures leading to a susceptible model on the novel environments. We expect that the proposed learning-based methods can potentially be more practical by not only just fitting well on the given data but also having the generalization ability.

초 록

소리의 종류 및 위치를 파악하는 것은 음향학 분야에서 가장 중요한 문제 중 하나이다. 특히 복잡한 건물 구조에서 기계적 결함 등으로 인한 소음원을 식별해야 할 경우 시각적 정보는 엄격히 차단되기 때문에 음향 정보에 의존 할 수 밖에 없다. 그러나 실제 복잡한 구조물은 철저히 계획된 실험이 아니기 때문에 소리의 전파가 이론을 따르기 않는다. 따라서 이 경우 음원의 종류 및 위치를 추정 하기 위해 고전적인 배열 처리 기술을 이용하는 기존의 접근 방식은 제한 될 수 밖에 없다. 따라서, 우리는 실제 건물에서 단일 마이크를 사용하여 음원의 종류 및 위치를 추정하는 학습 기반 접근법을 제안한다. 우리는 이 문제를 우리가 소리의 정확한 위치를 예측하는 동시에 미리 정의된 종류 중 하나로 분류하는 복합 분류 문제로 다루려고 한다. 음원의 종류는 원핫 인코딩 레이블로 지도 학습 프레임 워크에서 쉽게 분류 되지만, 가장 문제가 되는 부분은 훈련 중 보이지 않는 위치에서 나는 음원의 정확한 위치를 예측하는 것이다. 이러한 훈련 집합과 검증 집합의 잠재적 불일치를 해결하기 위해, 우리는 음원의 위치 추정 문제를 이전에 학습한 개념에서 새로운 개념을 지각하는 인간의 능력에서 영감을 받은 제로샷 학습 문제로 해결 하려한다. 우리는 각 분류군을 단순한 원핫 벡터로 레이블링 하는 대신 음성 데이터에서 특징 표현을 추출하고 음원의 종류 및 위치를 '종류/위치를 나타내는 속성'으로 벡터화한다. 이후, 우리는 음성에서 추출된 특징과 속성을 연결하기 위해 분류군에 따라 불변하는 함수를 검증된 생성 모델로 학습하여 해당 속성을 통해 훈련 중 보이는 클래서에서 보이지 않는 분류군으로 정보를 전이한다. 이때 생성 모델로 분류군 조건부 생성적 적대적 신경망을 이용한다. 우리가 제안한 방

법은 건물 내에서 수집된 실제 데이터셋인 소음 데이터셋, SNU-B36-EX에서 평가된다.

주요어: 음원 분류, 음원 위치 추정, 제로샷 학습, 생성적 적대 신경망
학번: 2017-28959