

TECHNISCHE UNIVERSITÄT MÜNCHEN
Lehrstuhl für Mensch-Maschine-Kommunikation

Automatic General Audio Signal Classification

Kun Qian

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik
der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Prof. Dr.-Ing. Wolfgang Utschick

Prüfer der Dissertation: 1. Prof. Dr.-Ing. habil. Björn W. Schuller
2. Prof. Dr.-Ing. Werner Hemmert

Die Dissertation wurde am 04.06.2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 08.11.2018 angenommen.

Acknowledgement

First of all, I am sincerely grateful to Prof. Björn Schuller, who is my supervisor and invited me to work with his group four years ago. Prof. Schuller always encourages me to do research on topics I am really interested in. Without his valuable suggestions and excellent supervisions, I could never achieve such work described in this thesis. The experience of working at Prof. Schuller's group in Munich, Passau and Augsburg is very happy and impressive for me.

Secondly, I would like to thank Prof. Gerhard Rigoll for allowing me to work at the Chair of Human-Machine Communication and providing me help both for my research and life at Munich. I am also grateful to Prof. TBA for his careful review and examination of this thesis, and Prof. TBA for chairing the examination board.

Thirdly, I am grateful to Prof. Satoshi Matsuoka from Tokyo Institute of Technology, Japan, and Assoc. Prof. Florian Metze from Carnegie Mellon University, USA, for their kind and helpful collaborative work contributions to some parts of this thesis. Many thanks are also given to Prof. Aimin Wang, Dr. Zixing Zhang, Dr. Jun Deng, Dr. Florian Eyben, Dr. Hesam Sagha, Dr. Anton Batliner, Dr. Martin Wöllmer, Dr. Xinzhou Xu, Mr. Vedhas Pandit, Mr. Maximilian Schmitt, Mr. Erik Marchi, Mr. Zijiang Yang, Ms. Jing Han, Ms. Zhao Ren, Ms. Yue Zhang, Mr. Peter Brand, Ms. Martina Römpf, Ms. Andrea Tonk, and Ms. Nadine Witek, for their technical and non-technical discussions during my very happy time in Germany. Specifically, I would like to thank Mr. Jian Guo, Mr. Tao Chen, Mr. Xiang Zha, Mr. Christoph Janott, Mr. Yang Zhang, Ms. Li Zhang, Ms. Hui Fang, and Mr. Zengjie Zhang, for their true help for both of research and life during my time of pursuing the doctoral degree. I also wish to thank Ms. Alice Baird, for her massive help in proof reading of my previous publications and this thesis.

Finally but most of all, this thesis is dedicated to my parents, Mr. Jinglei Qian and Ms. Jingping Zhang, who always give me power and support in my life.

Munich, May 2018

Kun Qian

Abstract

Automatic General Audio Signal Classification (AGASC), defined as machine listening based recognition of daily life audio signals rather than speech or music, is a very young field developing in recent years. Benefited from the state-of-the-art in the area of speech and music recognition, there are some standard methodologies (e. g., feature sets, classifiers, learning strategies) successfully applied into the area of AGASC. But more specific and robust models are still needed for advancing this area. This thesis proposes three typical tasks in AGASC, i. e., snore sound classification, bird sound classification, and acoustic scene classification, which represent the possible applications in healthcare, ecological monitoring, and public/home security surveillance, respectively.

The aim of this thesis is to facilitate the state-of-the-art in AGASC in the following: First, a comprehensive investigation on standard methodologies is given. To make the studies reproducible, the three databases used in this thesis are all publicly accessible. Second, some specifically-designed features, e. g., wavelet features, are presented and demonstrated to be very efficient for recognition of snore sounds and acoustic scenes. Furthermore, to reduce the human annotations on bird sound data, active learning is used. More precisely, a kernel based extreme learning machine is found to be superior to conventional support vector machines in the task of bird sound classification. Finally, a late fusion of multiple systems for acoustic scene classification in noisy environments is evaluated. All the experiments demonstrate the effectiveness of the methodologies proposed in this thesis.

Zusammenfassung

Die automatische Klassifikation von Audiosignalen (AKAS), welche sich mehr auf die maschinelle Erkennung von Audiosignalen aus dem täglichen Leben als auf Sprach- oder Musiksignale konzentriert, ist ein sehr junges Forschungsgebiet, das erst in den letzten Jahren entstanden ist. Vom Stand der Technik auf dem Gebiet der Sprach- und Musikerkennung profitieren einige Standardmethoden (z. B. akustische Merkmalsextraktion, Klassifikatoren, Lernstrategien), die erfolgreich im Bereich der AKAS Anwendung finden. Es bedarf jedoch noch spezifischerer und robusterer Modelle, um diesen Bereich weiter voranzutreiben. In dieser Arbeit werden drei typische Aufgaben zur AKAS untersucht, nämlich die Klassifikation von Schnarchgeräuschen, die Klassifikation von Vogelgesang und die Klassifikation akustischer Szenen, welche mögliche Anwendungen im Bereich der Gesundheitsvorsorge, der Ökologie und der öffentlichen Sicherheit darstellen.

Ziel dieser Arbeit ist es, den Stand der Technik in AKAS wie folgt zu fördern: Zunächst wird eine umfassende Untersuchung von etablierten Methoden des maschinellen Lernens durchgeführt. Um die Studien reproduktiv zu gestalten, sind die drei in dieser Arbeit verwendeten Datenbanken alle öffentlich zugänglich. Als nächstes werden einige speziell entworfene Audio-Merkmale, z. B. Wavelet-Merkmale, vorgestellt und es wird gezeigt, dass diese sehr effizient für die Erkennung von Schnarchgeräuschen und akustischen Szenen sind. Darüber hinaus wird aktives Lernen verwendet, um die notwendige Anzahl menschlicher Annotationen der Vogelgesang-Daten zu reduzieren. Es hat sich herausgestellt, dass eine kernelbasierte Extreme Learning Machine für die Klassifikation von Vogelgesang effizienter als eine herkömmliche Support Vector Machine ist. Schließlich wird eine späte Fusion mehrerer Systeme für die akustische Szenenklassifizierung in lauten Umgebungen evaluiert. Die beschriebenen Experimente belegen die Wirksamkeit der in dieser Arbeit vorgeschlagenen Methoden.

Relevant Publications

Refereed Journal Papers

- 2018 Kun Qian**, Christoph Janott, Zixing Zhang, Jun Deng, Clemens Heiser, Winfried Hohenhorst, Michael Herzog, Werner Hemmert and Björn Schuller, “Teaching Machines on Snoring: A Benchmark on Computer Audition for Snore Sound Excitation Localisation”, *Archives of Acoustics*, vol. 43, no. 3, pp. 465-475, 2018.
- 2018** Zhao Ren, **Kun Qian**, Zixing Zhang, Vedhas Pandit, Alice Baird and Björn Schuller, “Deep Scalogram Representations for Acoustic Scene Classification”, *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 3, pp. 662-669, 2018.
- 2018** Christoph Janott, Maximilian Schmitt, Yue Zhang, **Kun Qian**, Vedhas Pandit, Zixing Zhang, Clemens Heiser, Winfried Hohenhorst, Michael Herzog, Werner Hemmert and Björn Schuller, “Snoring Classified: The Munich Passau Snore Sound Corpus”, *Computers in Biology and Medicine*, vol. 94, pp. 106-118, 2018.
- 2017 Kun Qian**, Zixing Zhang, Alice Baird and Björn Schuller, “Active Learning for Bird Sound Classification via a Kernel-based Extreme Learning Machine”, *The Journal of the Acoustical Society of America*, vol. 142, no. 4, pp. 1796-1804, 2017.
- 2017 Kun Qian**, Christoph Janott, Vedhas Pandit, Zixing Zhang, Clemens Heiser, Winfried Hohenhorst, Michael Herzog, Werner Hemmert and Björn Schuller, “Classification of the Excitation Location of Snore Sounds in the Upper Airway by Acoustic Multi-Feature Analysis”, *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 8, pp. 1731-1741, 2017.
- 2017 Kun Qian**, Zixing Zhang, Alice Baird and Björn Schuller, “Active Learning for Bird Sounds Classification”, *Acta Acustica united with Acustica*, vol. 103, no. 3, pp. 361-364, 2017.

2017 Jian Guo, **Kun Qian**, Gongxuan Zhang, Huijie Xu and Björn Schuller, “Accelerating Biomedical Signal Processing Using GPU: A Case Study of Snore Sounds Feature Extraction”, *Interdisciplinary Sciences: Computational Life Sciences*, vol. 9, no. 4, pp. 550-555, 2017.

Refereed Conference Papers

2018 Zhao Ren, Nicholas Cummins, Vedhas Pandit, Jing Han, **Kun Qian** and Björn Schuller, “Learning Image-based Representations for Heart Sound Classification”, in *Proceedings of the 8th ACM Digital Health (DH)*, Lyon, France, ACM, pp.143-147, 2018.

2017 **Kun Qian**, Zhao Ren, Vedhas Pandit, Zijiang Yang, Zixing Zhang and Björn Schuller, “Wavelets Revisited for the Classification of Acoustic Scenes”, in *Proceedings of the Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, Munich, Germany, Tampere University of Technology, pp. 108-112, 2017.

2017 **Kun Qian**, Christoph Janott, Jun Deng, Clemens Heiser, Winfried Hohenhorst, Michael Herzog, Nicholas Cummins and Björn Schuller, “Snore Sound Recognition: On Wavelets and Classifiers from Deep Nets to Kernels”, in *Proceedings of the 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Jeju Island, Korea, IEEE, pp. 3737-3740, 2017.

2017 Tao Chen, **Kun Qian**, Antti Mutanen, Björn Schuller, Pertti Järventausta and Wencong Su, “Classification of Electricity Customer Groups Towards Individualized Price Scheme Design”, in *Proceedings of the 49th North American Power Symposium (NAPS)*, Morgantown, WV, USA, IEEE, pp. 1-4, 2017.

2017 Jian Guo, **Kun Qian**, Björn Schuller and Satoshi Matsuoka, “GPU-based Training of Autoencoders for Bird Sound Data Processing”, in *Proceedings of the 4th IEEE International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan)*, Taipei, Taiwan, IEEE, pp. 53-57, 2017.

2017 Zhao Ren, Vedhas Pandit, **Kun Qian**, Zijiang Yang, Zixing Zhang and Björn Schuller, “Deep Sequential Image Features for Acoustic Scene Classification”, in *Proceedings of the Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, Munich, Germany, Tampere University of Technology, pp. 113-117, 2017.

2017 Jun Deng, Nicholas Cummins, Maximilian Schmitt, **Kun Qian**, Fabien Ringeval and Björn Schuller, “Speech-based Diagnosis of Autism Spectrum

-
- Condition by Generative Adversarial Network Representations”, in *Proceedings of the 7th ACM Digital Health (DH)*, London, UK, ACM, pp. 53-57, 2017.
- 2017** Björn Schuller, Stefan Steidl, Anton Batliner, Erika Bergelson, Jarek Krajewski, Christoph Janott, Andrei Amatuni, Marisa Casillas, Amdanda Seidl, Melanie Soderstrom, S. Anne Warlaumont, Guillermo Hidalgo, Sebastian Schnieder, Clemens Heiser, Winfried Hohenhorst, Michael Herzog, Maximilian Schmitt, **Kun Qian**, Yue Zhang, George Trigeorgis, Panagiotis Tzirakis and Stefanos Zafeiriou, “The INTERSPEECH 2017 Computational Paralinguistics Challenge: Addressee, Cold & Snoring”, in *Proceedings of INTERSPEECH*, Stockholm, Sweden, ISCA, pp. 3442-3446, 2017.
- 2016** **Kun Qian**, Christoph Janott, Zixing Zhang, Clemens Heiser and Björn Schuller, “Wavelet Features for Classification of VOTE Snore Sounds”, in *Proceedings of the 41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, P. R. China, IEEE, pp. 221-225, 2016.
- 2016** Jian Guo, **Kun Qian**, Huijie Xu, Christoph Janott, Björn Schuller and Satoshi Matsuoka, “GPU-Based Fast Signal Processing for Large Amounts of Snore Sound Data”, in *Proceedings of the 5th IEEE Global Conference on Consumer Electronics (GCCE)*, Kyoto, Japan, IEEE, pp. 523-524, 2016.
- 2016** Maximilian Schmitt, Christoph Janott, Vedhas Pandit, **Kun Qian**, Clemens Heiser, Werner Hemmert and Björn Schuller, “A Bag-of-Audio-Words Approach for Snore Sounds Excitation Localisation”, in *Proceedings of the 12th ITG Conference on Speech Communication*, Paderborn, Germany, VDE, pp. 230-234, 2016.
- 2015** **Kun Qian**, Zixing Zhang, Fabien Ringeval and Björn Schuller, “Bird Sounds Classification by Large Scale Acoustic Features and Extreme Learning Machine”, in *Proceedings of the 3rd IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Orlando, FL, USA, IEEE, pp. 1317-1321, 2015.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Contributions	3
1.3	Structure of this thesis	4
2	Methodology	7
2.1	Acoustic Low-Level Descriptors	8
2.1.1	Formants	8
2.1.2	Spectral Frequency Features	10
2.1.3	Subband Energy Ratios	10
2.1.4	Mel-Frequency Cepstral Coefficients	11
2.1.5	COMPARE Feature Set Extracted by OPENSIMILE	11
2.1.6	Wavelet Features	11
2.2	Functionals and Bag-of-Audio-Words	15
2.2.1	Functionals	16
2.2.2	Bag-of-Audio-Words	18
2.3	Feature Normalisation	19
2.4	Classification	21
2.4.1	Classical Models	21
2.4.2	Deep Learning Models	25
2.4.3	Extreme Learning Models	33
2.5	Data Enrichment	36
2.5.1	Passive Learning	37
2.5.2	Active Learning	38
2.6	Late Fusion	40
2.7	Evaluation Metrics	41
2.7.1	Classification Evaluation	42
2.7.2	Significance Tests	42

3 Applications	45
3.1 Computer Audition for Snore Sound Excitation Localisation	46
3.1.1 Background	46
3.1.2 Munich Passau Snore Sound Corpus	48
3.1.3 Experimental Setup	50
3.1.4 Features and Classifiers for Snore Sound Classification	53
3.1.5 A Bag of Acoustic Features for Snore Sound Classification	56
3.1.6 Summary	60
3.2 Data Enrichment for Bird Sound Classification	64
3.2.1 Background	64
3.2.2 Museum für Naturkunde Berlin Bird Sound Database	66
3.2.3 Experimental Setup	67
3.2.4 Comparison of Passive Learning and Active Learning	68
3.2.5 Comparison of Robustness	69
3.2.6 Summary	72
3.3 Robust Systems for Acoustic Scene Classification	76
3.3.1 Background	76
3.3.2 DCASE 2017 Acoustic Scene Database	77
3.3.3 Experimental Setup	78
3.3.4 Acoustic Scene Classification in Clean Environments	79
3.3.5 Acoustic Scene Classification in Noisy Environments	81
3.3.6 Summary	81
4 Conclusion	89
4.1 Summary	89
4.2 Outlook	91
A Bird Species	93
List of Acronyms	95
List of Symbols	99
References	109

Introduction

1.1 Motivation

There is a long history for research in *automatic speech recognition* (ASR) [1], and *music information retrieval* (MIR) [2]. However, speech and music are only two of many types of sounds that can be heard in our daily life [3]. It is desirable within the audio research community to extend the frontiers of the state-of-the-art in ASR and MIR into more general sounds like *body acoustics*, *animal sounds*, *environmental sounds*, etc. The aim of this thesis is to facilitate research on combining *signal processing* and *machine learning* to find efficient and robust paradigms on features, classifiers and learning strategies for *automatic general audio signal classification* (AGASC) beyond speech and music. In this thesis, the classification tasks of three typical general audio signals, i. e., snore sound, bird sound, and acoustic scene, are proposed to evaluate the relevant methodologies applied to the area of healthcare, ecological monitoring, and public/home security surveillance, respectively:

1. *Snore Sound Classification*. *Snore sound* (SnS), carries information about the site and degree of obstruction in the upper airway of the subject [4]. There is an increasing need from the medicine community to find a less-invasive method, e. g., analysis of the snoring recordings, to understand the excitation localisation of SnS. In medical practice, it will be helpful for *Ear, Nose and Throat* (ENT) experts to plan a targeted surgery for both of the primary snorers (which are asymptomatic and do not have breathing interruptions during sleep), and the patients suffering from *obstructive sleep apnea* (OSA) [5], a chronic serious sleep disorder (when untreated, it can risk to stroke [6], hypertension [7], myocardial infarction [8], and even sudden death [9]) which affects approximately 13% of men and 6% of women in the U. S. population [10]. The reason is that, due to the multifactorial mechanisms of SnS generation, the surgical options for individual subjects can be manifold [11, 12]. In addition, *drug induced sleep endoscopy* (DISE) [13], as the current popular method

to identify the location and form of vibrations and obstructions in the upper airway, is time-consuming, costly and straining for the subjects. However, the studies focused on using audio information to localise the excitation of SnS are very limited [14]. The pilot work on using acoustic signal processing combined with machine learning for the recognition of SnS were published in [15–21], which showed promising and encouraging results. But, the subjects involved in these studies are extremely limited (less than 50). Further, the database the authors used are not publicly accessible, which makes the work difficult to be reproduced.

2. *Bird Sound Classification.* Recognition of bird species by their sounds can make it feasible to develop a long-term, non-human monitoring system for measuring the state of nature [22], tracking climate change [23], and assessing biodiversity within local ecosystems [24, 25]. This domain has attracted ornithologists, ecologists, and engineers in both *signal processing* and *machine learning*, to work towards applications for automatically classifying bird sound based only on the audio recordings throughout the past two decades. The mainstream of the previous relevant studies can be roughly divided into two directions, i. e., classification of bird sounds [26–41], or detection of syllables [42–45] from the bird sound audio recordings. Nevertheless, there are still few studies that focus on reducing the *human expert annotation* for the unlabelled bird sound data (segmented syllables, or continuous recordings) collected in the real-world. It was reported that, data collection, cleaning, and annotation alone will require approximately 80 % of the entire time needed in a typical data mining project [46]. More specifically, within the area of bird sound studies, there are large amounts of unlabelled audio recordings made in the field by ornithologists and amateurs, which bring forth a huge challenge for human annotators.
3. *Acoustic Scene Classification.* As a subfield of *computational auditory scene analysis* [47], *acoustic scene classification* (ASC) is defined as classification of the environment in which a recording has been made [48]. ASC is based on the assumption that, it is possible to use the general characterisations of a location to distinguish various acoustic scenes from one another by their general acoustic properties [48, 49]. Increasingly, there is large interest for ASC within the audio research community, which can stimulate areas like multimedia searching [50], smart mobile devices [51], and intelligent monitoring systems [52, 53]. More importantly, machine listening applications based on general environmental sound analysis can benefit applications in public/home security surveillance [54–56]. A recent paper reviewed the features and classifiers used for ASC task in [57]. The existing approaches regarding features include the use of *Mel-frequency cepstral coefficients* (MFCCs) [51, 58], his-

tograms of sounds [59], histogram of gradients learnt from time-frequency representations [60], and the time-frequency representations learnt by *nonnegative matrix factorization* (NMF) [61]. On the aspect of classifiers, *hidden Markov models* (HMMs) [59], *Gaussian mixture models* (GMMs) [58], and *support vector machines* (SVMs) [60, 62] were repeatedly investigated. More notably, the *deep learning* [63] methodologies are appearing as a mainstream direction for the ASC task [64–77]. However, the most investigated features are based on *Fourier* transformation [78], which cannot optimise the Heisenberg-alike time-frequency trade-off [79]. Besides, the existing studies are insufficient in investigation of combining the efficient features suitably for the ASC task in noisy conditions, which is the prerequisite in real product development.

In general, all the three tasks above are related to *signal processing* and *machine learning* as to methodologies like designing features, building learning models, selecting learning strategies (e. g., *supervised learning*, *unsupervised learning*, *semi-supervised learning*, *active learning*), etc. However, limited to the narrowed scope and field in each sub task, different emphasis is given on contributions of features for *snore sounds*, learning strategies for *bird sounds*, and combining models for *acoustic scenes*.

1.2 Contributions

To address the challenges listed above, this thesis makes contributions on the following aspects:

1. For *snore sound classification*, a comprehensive comparison on features and classifiers is provided by an open access SnS database, i. e., the *Munich Passau Snore Sound Corpus* (MPSSC) [80]. In particular, *wavelet features* are proposed and demonstrated to be efficient in classification of SnS. Furthermore, a method combining *wavelet features* [18] and *bag-of-audio-words* (BoAW) [19] is presented. It achieves the best result for SnS classification on the MPSSC database in this thesis.
2. For *bird sound classification*, *active learning* (AL) is used to reduce the need of human annotation for unlabelled data. In addition, a robustness comparisons between different AL algorithms, (i. e., *sparse-instance-based* AL (SI-AL) and *least-confidence-score-based* AL (LCS-AL)) and their performances when using two popular classifiers (i. e., *support vector machine* (SVM) [81] and *kernel-based extreme learning machine* (KELM)) [82] is investigated. The algorithms previously shown in [83] are successively extended from a binary classification problem to multi-class classification problem. Moreover, it is found that,

changing to a new classifier, e. g., KELM, can considerably improve the performance of AL for bird sound classification. All the experiments are evaluated by a public bird sound database, i. e., the *Museum für Naturkunde Berlin* (MNB) bird sound database.

3. For *acoustic scene classification*, the effectiveness of *wavelets* [84] and a late fusion system of multiple features are investigated. The methodologies are evaluated on the dataset *Detection and Classification of Acoustic Scenes and Events* (DCASE) 2017 [48], an active and influential challenge among the emerging evaluation campaigns and datasets in the area of environmental sound classification and detection, which had also been successfully held in 2013 [85] and 2016 [86]. Furthermore, comparisons of different feature sets in noisy environments are also given. As to classifiers, SVM, GRNN and BGRNN (*gated recurrent neural network* [87], *bidirectional gated recurrent neural network* [87, 88]) are selected and compared in this thesis for the ASC task.

Generally, some other contributions for AGASC from this thesis can also be briefly concluded as: Firstly, to make the work described in this thesis reproducible and sustainable, all the databases used are publicly accessible. Secondly, this thesis gives standard benchmarks on features, classifiers, learning strategies, and evaluation metrics for AGASC. Finally, the presented methodologies and implementations can be extended and applied into other relevant work on other general sound classification tasks, e. g., insect sounds, heart beat sounds, mechanical noises, etc.

1.3 Structure of this thesis

The rest of this thesis is organised as follows: *Chapter 2* will describe the methodologies in details. The experimental results and discussions are given in *Chapter 3*. Finally, *Chapter 4* gives a summary of the current work with the limitations, and an outlook for the future work. In more detail:

Chapter 2 describes a set of methods used for dealing with the challenges proposed in Section 1.1. To be more specific, a comprehensive study on features and classifiers are investigated for *snore sound classification*. For *bird sound classification*, active learning strategies are used to considerably reduce the human annotations. For *acoustic scene classification*, the fused models are demonstrated to be more efficient and robust than a single model.

Chapter 3 shows the implementations of the methods presented in Chapter 2. All the experimental results are based on the evaluations on publicly accessible databases. For each sub task mentioned in Section 1.1, different methods are used to address the corresponding challenges. A brief summary will also be provided to each application.

Chapter 4 summarizes and concludes the current work done in this thesis. The limitations are also discussed to point out some possible directions of future work, which are given in an outlook.

Methodology

This chapter introduces the methods used in this thesis. Typically, there are two main stages for building a system for general audio signal classification, i. e., *feature extraction* and *model learning*. Figure 2.1 briefly gives an overview of the system which plays the main role in this thesis. More precisely, the whole system can be referred to as two modules, i. e., the *front-end* and the *back-end* (refer to [89]). In the front-end, the first goal is to enhance the input audio signal that might be distorted by interfering noise, environmental noise, recording equipment noise, and reverberation. Many signal processing methods, e. g., adaptive filtering [90], spectral normalisation and subtraction [91], or beamforming [92], can be employed to overcome the effects of noise. Then, a stage of *audio activity detection* (AAD) will be used to detect the *events* for further analysis. For instance, in an overnight audio recording of snore sounds, the snore related events will be firstly detected and segmented [17]. Subsequently, the detected audio signal will be sent to the feature extraction phase, in which a series of acoustical *low-level descriptors* (LLDs) are computed at a frame-level in a predefined window (with overlap). In this thesis, for Fourier transformation based features, a Hamming window [78] is used before extracting LLDs. Further, a summarisation of these LLDs over a segment (e. g., an instance) can be achieved by *functionals* or a *bag-of-audio-words* approach. To reduce the feature dimensionality, a feature selection stage is usually added. In the back-end, extracted features are fed into a machine learning model for its training and tuning. The parameters of the classification models can be tuned and optimised by train and development data sets, which have the labelled targets. The final optimised model can be evaluated by predicting the labels of the test set (labels are unseen).

This thesis mainly focuses on the phase of feature extraction and model learning (as Figure 2.1 shows), which will be introduced in the following sections.

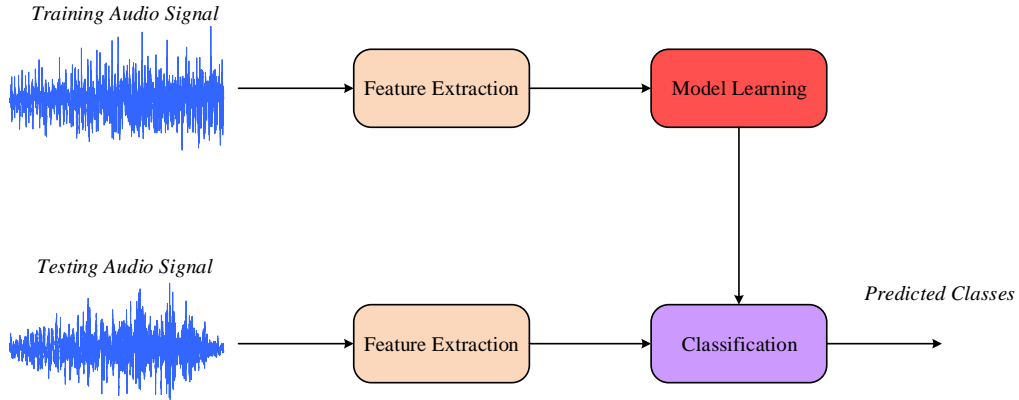


Figure 2.1: Overview of a general audio signal classification system.

2.1 Acoustic Low-Level Descriptors

Acoustic *low-level descriptors* (LLDs) are raw representations extracted from a short-time frame of the analysed audio signal. Extracting efficient LLDs suitably reflecting the inherited characters of the analysed audio signal is an important phase in the front-end module of building an audio classification system. Typically, for speech recognition, *prosodic* features, e.g., intensity, fundamental frequency F0, voicing probability, formants, and *cepstral* features, e.g., *linear prediction cepstral coefficients* (LPCCs), *Mel-frequency cepstral coefficients* (MFCCs) are often used. In the past decades, those acoustical LLDs have been demonstrated to be efficient and robust in a tremendous amount of speech recognition related tasks. However, directly using the conventional acoustic LLDs to train a model for a general audio signal classification task might not always be successful or robust enough. For instance, a snore sound is extremely different to a speech signal (see Figure 2.2) not only in the time waveform, but also in the spectral distribution. In a pilot study [18] of this thesis, *wavelet* features were found to be superior to formants and MFCCs, which are widely used features in snore sound analysis. In addition, the conventional frame length (usually about 10-30 ms) for extracting LLDs from speech might not be suitable for other audio signals. An empirical study [93] showed that the frame length and overlap for extracting LLDs from a snore sound can effect the final classification of the trained models. In this section, a series of acoustical LLDs used in this thesis will be introduced and described as follows.

2.1.1 Formants

As defined in [94], formants are the spectral peaks of the sound spectrum. In previous studies on the analysis of SnS, formants were widely investigated [95–97]. For otolaryngologists, formants can reflect the anatomical structure of the upper

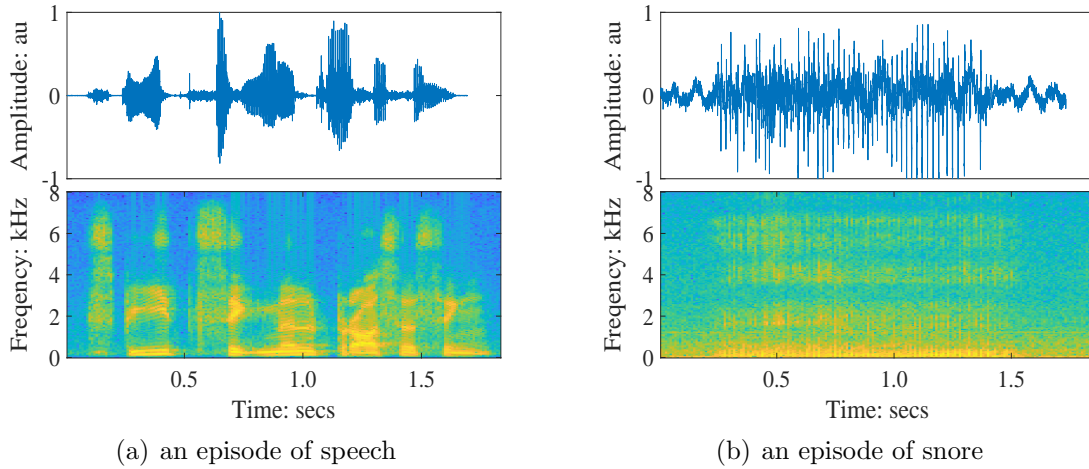


Figure 2.2: Examples of waveforms (top row, normalised) and spectrograms (bottom row) for an episode of speech and snore. au: arbitrary unit.

airway. The first formant (F1) is thought to be associated with the tongue height; the second formant (F2) is considered to be related to the degree of tongue advancement from its neutral position; and the third formant (F3) is regarded as a factor to indicate the amount of lip rounding [98]. In this thesis, linear predictive coding (LPC) is used to estimate the formant frequencies [99–101]. Given a set of predictor coefficients a_0, a_1, \dots, a_p ($a_0 = 1$), the z -domain transfer function of the filter (all-pole *autoregressive* (AR) modeling [102]) can be expressed as [99]:

$$H(z) = \frac{1}{1 - \sum_{i=1}^p a_i z^{-i}}, \quad (2.1)$$

where the LPC coefficients will be determined via the Yule-Walker autoregressive method along with the Levinson-Durbin recursive procedure [101]. Then, the i -th formant frequency F_i can be estimated as:

$$F_i = \frac{F_s}{2\pi} \left| \arctan \left(\frac{\text{Im}(r_i)}{\text{Re}(r_i)} \right) \right|, \quad (2.2)$$

where F_s is the sampling frequency of the audio signal, and $\text{Im}(r_i)$ and $\text{Re}(r_i)$ are the imaginary and real part for the i -th root (r_i) of the model above, respectively. We need to note that, the roots only with the positive imaginary parts are retained when calculating the formant frequencies due to the fact that LPC coefficients are real-valued and the roots are symmetric on the imaginary axis.

2.1.2 Spectral Frequency Features

It was found that, the spectrum of SnS can carry important information on distinguishing the snore sites [103]. In addition, the spectral frequency features were proven to be efficient at both diagnosis of OSA [104], and classification of snore related signals [16, 17, 21]. In this thesis, the peak frequency (F_{peak}), the centre frequency (F_{centre}), the mean frequency (F_{mean}), and the mean frequency in each subband spectrum ($F_{mean-sub}$) are extracted from SnS and combined together as *spectral frequency features* (SFFs).

The peak frequency is defined as [103]:

$$F_{peak} : s.t. X_{F_{peak}} = \max\{X_{f_n}\}, n = 1, 2, \dots, \frac{N}{2} + 1, \quad (2.3)$$

where X_{f_n} is the absolute value of the single-sided amplitude by *fast Fourier transformation* (FFT) [78] of the audio signal at the n -th point. The FFT is performed at N points (N is equal to the length of the audio signal in this thesis).

The centre frequency is defined as [103]:

$$F_{centre} : s.t. \sum_{f_n=0}^{F_{centre}} X_{f_n} = \sum_{f_n=F_{centre}}^{F_h} X_{f_n}, \quad (2.4)$$

where F_h is the highest frequency of the spectrum of the audio signal, i. e., $\frac{F_s}{2}$ by Nyquist sampling theorem [78].

The mean frequency is defined as [21]:

$$F_{mean} = \frac{\sum_{f_n=0}^{F_h} f_n X_{f_n}}{\sum_{f_n=0}^{F_h} X_{f_n}}. \quad (2.5)$$

The mean frequency in each subband spectrum is defined as [21]:

$$F_{mean-sub(m)} = \frac{\sum_{f_n=(m-1)F_b}^{mF_b} f_n X_{f_n}}{\sum_{f_n=(m-1)F_b}^{mF_b} X_{f_n}}, m = 1, 2, \dots, \lfloor \frac{F_h}{F_b} \rfloor, \quad (2.6)$$

where F_b is a frequency to set subband of the whole spectrum.

2.1.3 Subband Energy Ratios

The energy distributions over the frequency spectrum might differ between different types of SnS. In early studies, *subband energy ratios* (SERs) were found efficient

in snore/nonsnore classification [105, 106]. In this thesis, as a feature set, SERs is defined as:

$$SER_{(m)} = \frac{\sum_{f_n=(m-1)F_b}^{mF_b} (X_{f_n})^2}{\sum_{f_n=0}^{F_h} (X_{f_n})^2}, m = 1, 2, \dots, \lfloor \frac{F_h}{F_b} \rfloor. \quad (2.7)$$

2.1.4 Mel-Frequency Cepstral Coefficients

Mel-Frequency cepstral coefficients (MFCCs) are one of the most popular features used in speech recognition. The real scale frequency f (in Hz) can be mapped by triangular overlapping filters onto the a Mel-scale $f_{(Mel)}$ (in Mels) as [107]:

$$f_{(Mel)} = 2595 \log_{10} \left(1 + \frac{f}{700} \right). \quad (2.8)$$

The calculations of MFCCs take the non-linear frequency perception of the human ear into account, in which, the frequencies between 0 Hz and 1 kHz are linearly approximated and a logarithmic scale will be applied for frequencies beyond 1 kHz. Details on calculation of MFCCs can be found in [107].

2.1.5 ComParE Feature Set Extracted by openSMILE

The COMPARE feature set, firstly proposed in the INTERSPEECH 2013 Computational Paralinguistics Challenge (COMPARE) [108], is a set of standard widely-used temporal and spectral acoustic features. This feature set extracted by the OPENSMILE toolkit [109, 110] has been successfully used in various general audio signal classification tasks (e. g., snore sound [111], bird sound [112], or acoustic scene [113]), and continuously adopted to build the baseline system for the recent computational paralinguistics challenges [108, 111, 114–116]. An overview of the 65 LLDs in the ComParE feature set is provided in Table 2.1. As the usage of the COMPARE feature set for AGASC is not the main direction of this thesis, the detailed methodologies for extracting each kind of LLD can be found in [117].

2.1.6 Wavelet Features

The aforementioned LLDs are mainly extracted via the *short-time Fourier transformation* (STFT) [118]. It should be noted that, STFT has been successfully applied to time-frequency analysis for decades, e. g. speech analysis. However, it does not optimise the Heisenberg-like time-frequency trade-off [79], i. e., a good resolution of the analysed signal cannot be achieved by STFT simultaneously in the time and the frequency domain. It can be simply explained that, for a certain length (constant) of the analysed signal, increasing the frequency resolution will cause a larger

Table 2.1: Overview of low-level descriptors (LLDs) in the COMPARE feature set. RMSE: Root Mean Square Energy; ZCR: Zero-Crossing Rate; RASTA: Representations Relative Spectra; RoP: Roll-off Point; SHS: Subharmonic Summation; HNR: Harmonics to Noise Ratio. The source of the table can be found in [117].

Group A: (59)
Loudness, Modulation loudness, RMSE, ZCR, MFCC 1–14 RASTA auditory bands 1–26, Energy 250–650 Hz, Energy 1–4 kHz Spectral RoP .25, .50, .75, .90, Spectral flux, Spectral entropy, Spectral variance, Spectral slope, Spectral skewness and kurtosis Spectral harmonicity, Spectral sharpness (auditory), Spectral centroid (linear)
Group B: (6)
F0 via SHS, Probability of voicing, Jitter (local and delta), Shimmer log HNR (time domain)

window size (increasing N in one window) and therefore a reduction in time resolution (decreasing the number of windows), and vice versa (see Figure 2.3). Thus, STFT is not suitable for analysing signals which include structures having different time-frequency resolutions whereas *wavelets* can address this issue by changing the time-frequency resolution [84, 119].

In this thesis, *wavelet transformation* (WT) is chosen as *discrete wavelet transformation* (DWT), which is usually used to extract multi-resolution representations from the analysed signal. Figure 2.4 shows the capacity of multi-resolution analysis by wavelets. Generally, *wavelets* are defined by the *wavelet function* $\psi_{j,k}(n')$ and the *scaling function* $\phi_{j,k}(n')$, respectively. In the following, n' denotes the index of value in the functions. The two aforementioned functions are defined as [84]:

$$\psi_{j,k}(n') = \frac{1}{\sqrt{2^j}} \psi\left(\frac{n' - 2^j k}{2^j}\right), \quad (2.9)$$

$$\phi_{j,k}(n') = \frac{1}{\sqrt{2^j}} \phi\left(\frac{n' - 2^j k}{2^j}\right), \quad (2.10)$$

where j (nonnegative integer) and k (nonnegative integer) denote the *scale* and the *index* of a subband within the *scale*, respectively. The value 2^j , known as the ‘*scaling parameter*’, measures the scaling. The ‘*translation parameter*’, i.e., the value of $2^j k$, reveals the time location of the wavelet. The mechanism of WT is to decompose the signal from the original space $\mathbf{V}_{j,k}$ into two orthogonal subspaces, i.e., an *approximation* space $\mathbf{V}_{j+1,2k}$, and a *detail* space $\mathbf{V}_{j+1,2k+1}$ [84]. This process can be done by dividing the orthogonal basis $\{\phi_j(n' - 2^j k)\}$ of $\mathbf{V}_{j,k}$ into two new

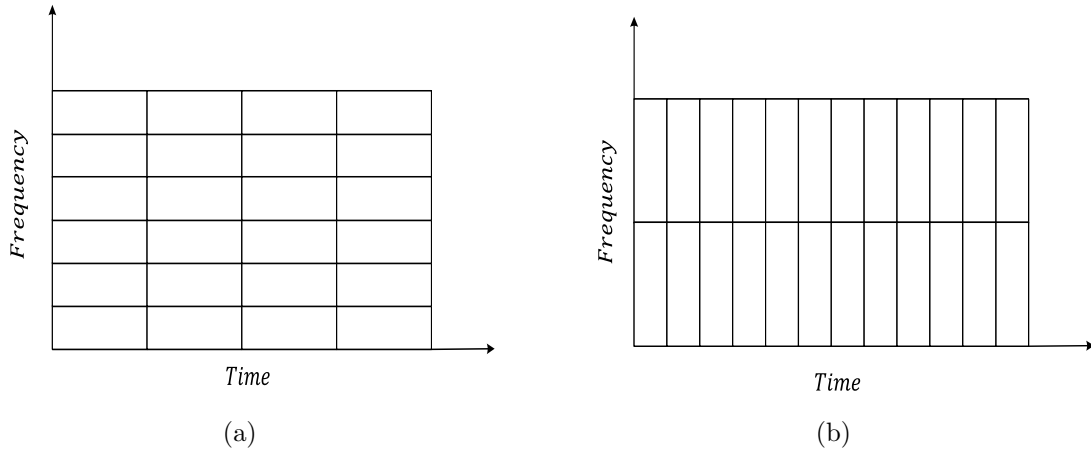


Figure 2.3: Time-frequency resolution by STFT. The left (a) has a better resolution in frequency but a worse resolution in time than the right (b).

orthogonal bases, i. e., $\{\phi_{j+1}(n' - 2^{j+1}k)\}$ of $\mathbf{V}_{j+1,2k}$, and $\{\psi_{j+1}(n' - 2^{j+1}k)\}$ of $\mathbf{V}_{j+1,2k+1}$ [84].

A vector of *approximation* coefficients and a vector of *detail* coefficients can be obtained via the aforementioned orthogonal wavelet decomposition procedure [84]. Nevertheless, this analysis is based on a coarser scale due to the information lost between two successive approximations in the *detail* coefficients, i. e., the WT will not decompose the *detail* coefficients into the subsequent decomposition levels (see Figure 2.5(a)). In contrast, the *wavelet packet transformation* (WPT) [120,121], not only decomposes the *approximation* coefficients, but also the *detail* coefficients into the subsequent decomposition levels, which in result produces a complete binary tree (see Figure 2.5(b)).

In the following, two wavelet-based features, i. e., *wavelet transform energy* (WTE) and *wavelet packet transform energy* (WPTE) will be described, respectively. The two kinds of wavelet features were originally designed by Khushaba *et al.* [122,123] and the scripts for extracting the features are accessible¹. Additionally, an early fusion of WTE and WPTE, called *wavelet feature energy* (WEF), will be given as another feature set, which was successfully applied to SnS classification [18,21] and the ASC task [113].

¹<https://www.rami-khushaba.com/matlab-code.html>

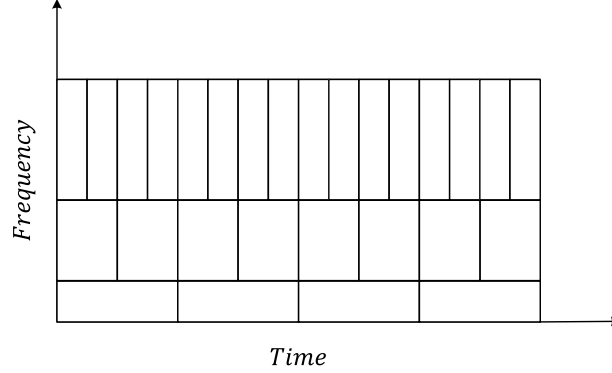


Figure 2.4: Time-frequency resolution by wavelet analysis. Good frequency resolution can be achieved in low frequencies, whereas good time resolution can be achieved in high frequencies.

Wavelet Transform Energy

The WTE feature set contains the variations of the WT coefficients' energy percentage. At each decomposition level, a vector of energy percentage is defined as:

$$\mathbf{E}_j = \frac{(\mathbf{w}_j)^2}{\sum_{j=1}^{J_{max}} (\mathbf{w}_j)^2} \times 100, \quad (2.11)$$

where \mathbf{w}_j is the vector of coefficients generated by WT at the j -th decomposition level. The parameter J_{max} is the maximum level for wavelet decomposition. Subsequently, the *mean*, the *variance*, the *waveform length* (the sum of squared differences between the adjacent elements of a vector), and the *entropy* [124] are calculated from the vector \mathbf{E}_j . In total, there are vectors representing the *detail* coefficients' energy percentage from level 1 to level J_{max} and the *approximation* coefficients' energy percentage at level J_{max} will be used to generate the LLDs of the WTE, which in result leads to a dimension of $4 \times (J_{max} + 1)$.

Wavelet Packet Transform Energy

The WPTE feature set contains the normalised filter bank energy (added with the natural logarithmic operator), which is defined as [123]:

$$\hat{E}_{j,k} = \log \sqrt{\frac{\sum_{n''=1}^{N_{j,k}} (\mathbf{w}_{j,k,n''})^2}{N_{j,k}}}, \quad (2.12)$$

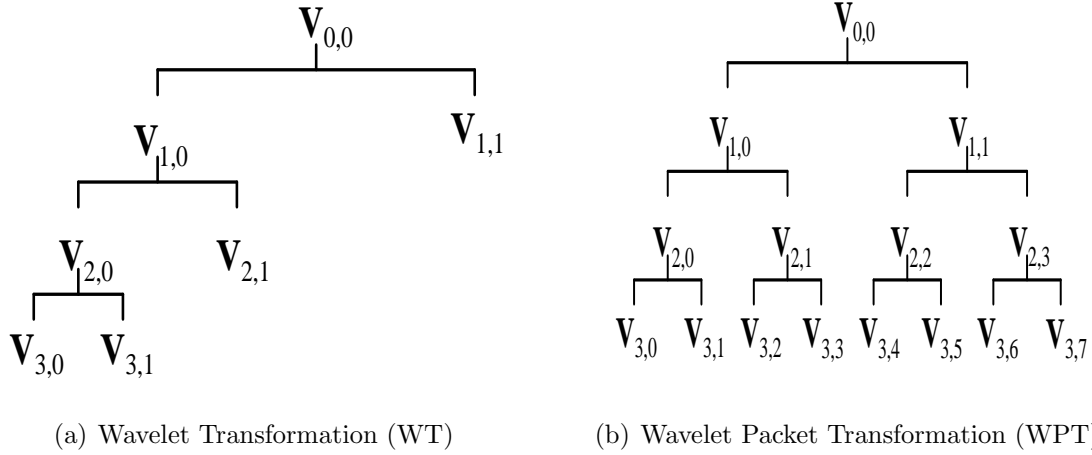


Figure 2.5: Example of 3-level decomposition tree-structured subspaces by WT and WPT. Compared with WT, WPT not only decomposes the *approximation* coefficients, but also the *detail* coefficients in the subsequent decomposition levels.

where $\mathbf{w}_{j,k,n''}$ (n'' denotes the index of the coefficients) represents the vector of coefficients calculated by WPT from the analysed signal at the subspace $\mathbf{V}_{j,k}$. $N_{j,k}$ is the total number of coefficients in k -th subband at j -th decomposition level. The scale of k is $0, 1, 2, \dots, 2^j - 1$. In the j -th decomposition level, there will be 2^j LLDs extracted as the WPTE feature. The $\hat{E}_{j,k}$ will be extracted from the level 0 (i. e., the original analysed signal) to the level J_{max} , which results in a dimension of $2^{J_{max}+1} - 1$.

Wavelet Energy Feature

In a previous study [93], an *early fusion* of the two aforementioned wavelet features was found to be efficient and even better than merely using WTE or WPTE. However, this is usually dependent on application. In this thesis, both WTE, WPTE and the early fusion of the two feature sets will be investigated. The early fusion of WTE and WPTE is referred to as the *wavelet energy feature* (WEF) in the following.

2.2 Functionals and Bag-of-Audio-Words

In this section, two approaches applied to LLDs extracted from one instance will be described, i. e., *functionals* and *bag-of-audio-words* (BoAW). Both of the two approaches can summarise the statistical information of LLDs over a given time period. In addition, after the processing of functionals or within the BoAW approach, the frame-based LLDs can be fixed into a single vector independent of the length of the

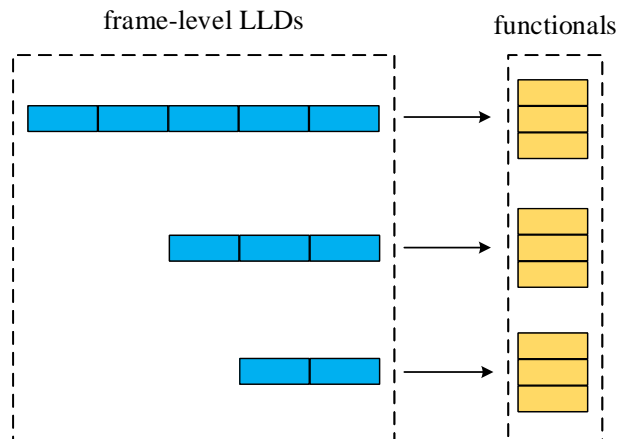


Figure 2.6: Mechanism of functionals. The frame-level LLDs from instances (with varied length) can be mapped into fixed dimension vectors (with the same length) by functionals.

original instance, which makes it applicable for static models (e. g., SVMs, see Section 2.4.1) to learn the patterns in the analysed audio signals. As a well-developed method, functionals are widely used in many speech/audio related applications and became a standard configuration in the large scale feature extraction open source toolkit `OPENSILE` [109, 110]. Recently, the BoAW approach has become popular for its comparable or even superior performance to functionals [19, 125] and can be implemented via the open source toolkit `OPENXBOW` [126].

2.2.1 Functionals

In the analysis of general audio signals, the change of low-level features over a given period of time can carry important information for a further model building step. The *supera-segmental* features can summarise the information over a meaningful unit of time [117]. As mentioned in [117], a straightforward approach is to stack all the LLD vectors to a single, large vector, in which all the information from the original features can be contained. However, if the number and dimensionality of the LLD vector is large, specifically, when the analysed audio signal is long, it will dramatically increase the computational cost and test time. Instead, *functionals* can be applied to the time series of LLDs (frame-level LLDs) [117], which results in a single, fixed dimension vector independent of the length of the input. The mechanism of functionals is to map the time series to a scalar value per applied functional. Figure 2.6 briefly shows the mechanism of functionals applied to frame-level LLDs.

Typical functionals include the arithmetic mean, standard deviation, and ex-

tremes (minimum value, maximum value). More advanced functionals, e. g., moments, percentiles, regression, can be found in [117]. In this thesis, four kinds of functionals, i. e., the *maximum value*, the *minimum value*, the *arithmetic mean*, and the *linear regression offset*, which were found efficient in previous studies [20,21,113] are considered. Let $y = x(n), n = 1, 2, \dots, N$ denote a general time series (the LLD values in a given time period), the maximum, minimum, and arithmetic mean value can be expressed as:

$$x_{max} = \max\{x(1), x(2), \dots, x(N)\}, \quad (2.13a)$$

$$x_{min} = \min\{x(1), x(2), \dots, x(N)\}, \quad (2.13b)$$

$$x_{mean} = \frac{1}{N} \sum_{n=1}^N x(n). \quad (2.13c)$$

For linear regression, the goal is to approximate a line ($\hat{y} = an+b$) that has minimised quadratic error between the line (\hat{y}) and the actual series (y). The quadratic error can be written as:

$$\begin{aligned} \hat{e}^2 &= \sum_{n=1}^N (y - \hat{y})^2 = \sum_{n=1}^N (x(n) - an - b)^2 \\ &= \sum_{n=1}^N (x(n)^2 - 2anx(n) - 2bx(n) + 2abn + a^2n^2 + b^2), \end{aligned} \quad (2.14)$$

where a is the slope, and b is the offset. To minimise \hat{e}^2 , the following differential equations can be expressed as (can be referred to [117]):

$$\frac{\partial \hat{e}^2}{\partial a} = \sum_{n=1}^N (-2nx(n) + 2bn + 2an^2) = 0, \quad (2.15a)$$

$$\frac{\partial \hat{e}^2}{\partial b} = \sum_{n=1}^N (-2x(n) + 2an + 2b) = 0, \quad (2.15b)$$

which can re-written as:

$$-\sum_{n=1}^N nx(n) + b \sum_{n=1}^N n + a \sum_{n=1}^N n^2 = 0, \quad (2.16a)$$

$$-\sum_{n=1}^N x(n) + a \sum_{n=1}^N n + Nb = 0. \quad (2.16b)$$

Then, the solutions for a and b can be yielded as:

$$a = \frac{N \sum_{n=1}^N nx(n) - \sum_{n=1}^N n \sum_{n=1}^N x(n)}{N \sum_{n=1}^N n^2 - \left(\sum_{n=1}^N n\right)^2}, \quad (2.17a)$$

$$b = \frac{\sum_{n=1}^N x(n) \sum_{n=1}^N n^2 - \sum_{n=1}^N n \sum_{n=1}^N nx(n)}{N \sum_{n=1}^N n^2 - \left(\sum_{n=1}^N n\right)^2}. \quad (2.17b)$$

In the previous study [21], the offset (b) was found to be more efficient than the slope (a). Therefore, in this thesis, the offset is included as a complementary functional to aforementioned x_{max} , x_{min} , and x_{mean} . The four functionals are applied to the LLDs of Formants (Section 2.1.1), SFFs (Section 2.1.2), SERs (Section 2.1.3), MFCCs (Section 2.1.4), and wavelet features (Section 2.1.6). The functionals used for the COMPARE feature set (Section 2.1.5) are listed in Table 2.2. More detailed definitions and calculations of functionals can be found in [117].

2.2.2 Bag-of-Audio-Words

The *bag-of-audio-words* (BoAW) approach has originated from the *bag-of-words* (BoW) principle, which can be referred to as the early description in [127]. The BoW method is applicable in *natural language processing* [128], and being adopted by the research community of *computer vision* [129,130]. Recently, the BoW method has been widely used in audio classification tasks (referred to the term as BoAW) like *acoustic event classification* [131–134], *multimedia event detection* [135–137], *speech emotion recognition* [125,138], and *health care* [19,139].

In the BoAW approach, term frequency histograms are generated from the acoustic LLDs. Compared to the BoW approach, where text documents are represented as word histograms, the numerical LLDs extracted from the audio signal need to undergo a *vector quantisation* (VQ) step first. The VQ is done employing a *codebook* of template LLDs which is previously learnt from a certain amount of training data. Although the codebook generation usually employs *k-means clustering* [135] (see Algorithm 1), similar results can be achieved using a *random sampling* of the LLDs [140], where the sampling follows the initialisation step of e.g., *k-means++ clustering* [141] (see Algorithm 2, which betters the initialisation step of *k-means*), i.e., far-off LLDs are prioritised. Instead of assigning each LLD to only the most similar word in the codebook, the N_a words with the lowest *Euclidean* distance can be considered, which usually results in an improved robustness of the approach [125]. In the resulting histogram, the logarithm (with a bias of 1) is then taken from the word frequencies, in order to compress the range of values. The whole process of

Table 2.2: Overview of functionals applied to LLDs in the COMPARE feature set. The functionals marked with A and B are only applied to Group A or B (see Table 2.1) LLDs and the *delta* LLDs(referred to [117]), respectively. The functionals marked with ζ or η are not or only applied to the delta LLDs, respectively. The source of the table can be found in [117].

<i>Functionals</i>
Arithmetic $^{A\zeta,B}$ or positive arithmetic $^{A\eta,B}$ mean
Root-quadratic mean, flatness
Standard deviation, skewness, kurtosis, quartiles 1–3
Inter-quartile ranges 1–2, 2–3, 1–3, 99-th and 1-st percentile, range of these
Relative position of max. and min. value, Range (maximum to minimum value)
Linear regression slope $^{A\zeta,B}$, offset $^{A\zeta,B}$, Linear regression quadratic error $^{A\zeta,B}$
Quadratic regression coeff. $^{A\zeta,B}$, Quadratic regression quadratic error $^{A\zeta,B}$
Temporal centroid $^{A\zeta,B}$, Peak mean value A and distance to arithmetic mean A
Mean A and std. dev. A of peak to peak distances
Peak and valley range A (absolute and relative)
Peak-valley-peak slopes mean A and standard deviation A
Segment length mean A , min. A , max. A , standard deviation A
Up-level time 25 %, 50 %, 75 %, 90 %
Rise time, left curvature time, Linear Prediction gain and coefficients 1–5

BoAW generation is exemplified in Figure 2.7. In order to reduce the effect of different magnitudes between the LLDs, they are subject to standardisation. Accordingly, also the resulting term frequency histograms are standardised before they are fed into a classifier.

The *codebook size*, i. e., C_s , and the *number of assignments*, i. e., N_a are crucial parameters in the BoAW approach and need to be optimised [19, 125]. However, there is no general best practice regarding the aforementioned parameters. More precisely, as indicated in [125], the optimum C_s depends not only on the number and type of LLDs, but also the task. In this thesis, the C_s and N_a will be tuned empirically to be optimised in initial experiments. Besides, the open source toolkit OPENXBOW [126] will be used to implement the whole BoAW approach, which can make the experiments reproducible.

2.3 Feature Normalisation

The extracted features usually have their own physical meaning, which in turn have their corresponding specific units. Besides, due to the variable circumstances of the

Algorithm 1: k -means Clustering

- 1 Randomly choose k initial centres: $C = \{c_1, c_2, \dots, c_k\}$.
 - 2 **repeat**
 - 3 Set the cluster \mathcal{S}_i as a set of points in the data set \mathcal{X} that are closer to c_i than to c_j ($\forall i \neq j, i, j \in \{1, 2, \dots, k\}$).
 - 4 Set c_i as the centre in \mathcal{S}_i : $c_i = \frac{1}{|\mathcal{S}_i|} \sum_{x \in \mathcal{S}_i} x$ $i \in \{1, 2, \dots, k\}$.
 - 5 **until** the set of centres C no longer changes.
-

Algorithm 2: k -means++ Clustering

- 1 Randomly choose one centre c_1 from the data set \mathcal{X} .
 - 2 **repeat**
 - 3 Choose a new centre c_i : Choosing $x \in \mathcal{X}$ with a probability $\frac{D(x)^2}{\sum_{x \in \mathcal{X}} D(x)^2}$, where $D(x)$ denotes the shortest distance from x to the closest centre that has already been chosen.
 - 4 **until** k centres have been chosen.
 - 5 Proceed as the standard k -means clustering.
-

audio environments, subjects and recording equipment, a normalisation technique is needed to unify the feature values within a small specified range. Particularly, for training neural networks, feature sets should be normalised before building the models to speed up the the learning phase [142]. Given a feature vector \mathbf{x} , simply expressed as $x(n)$ for $n = 1, 2, \dots, N$ (N represents the number of values in this vector), there are two main methods [89, 117] to normalise \mathbf{x} .

Min-max normalisation scales the feature values into a predefined interval. In this method, the normalised feature vector \mathbf{x}' has values as:

$$x'(n) = \frac{x(n) - \mu_{\mathbf{x}}}{x_{max} - x_{min}} \cdot (b - a) + a, \quad (2.18)$$

where the value a and b define the interval $[a, b]$ by linear scaling. The scalar $\mu_{\mathbf{x}}$, x_{max} , x_{min} represents the arithmetic mean, maximum, and minimum values calculated from the vector \mathbf{x} . Usually, two scales are used, i. e., $[0, 1]$ and $[-1, 1]$. As indicated in [117], this method is vulnerable to single outliers, which limits its application in realistic conditions.

Standardisation (also called z-score normalisation) forces the feature values to have an arithmetic mean of zero and a variance of one. The normalised values can be expressed as:

$$x'(n) = \frac{x(n) - \mu_{\mathbf{x}}}{\sigma_{\mathbf{x}}}, \quad (2.19)$$

where $\sigma_{\mathbf{x}}$ is the standard deviation of the vector \mathbf{x} . As indicated in [89], standardisation is more robust to outliers than min-max normalisation.

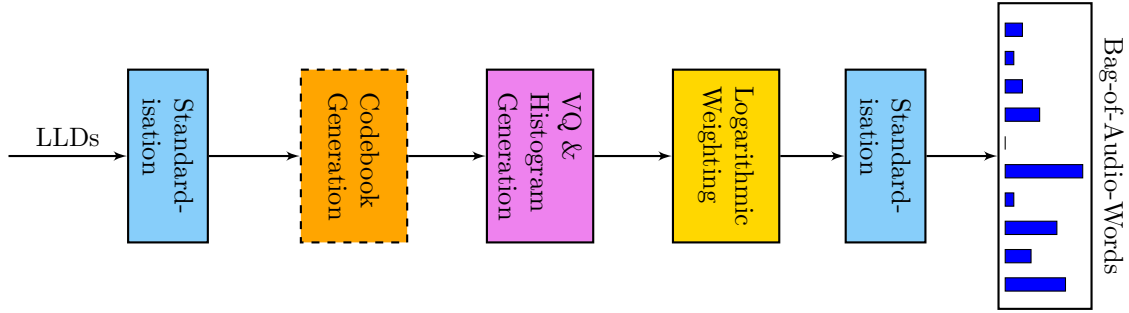


Figure 2.7: Diagram of the bag-of-audio-words generation process. The *codebook generation* is only performed in the training phase.

In this thesis, features are normalised via the standardisation method except an according statement is given. It should be noted that, the parameters $\{\mu_{\mathbf{x}}, x_{min}, x_{max}, \sigma_{\mathbf{x}}\}$ of one certain feature vector are extracted from the *train set* and applied to its counterpart in the *development* and *test* sets.

2.4 Classification

Given an instance (\mathbf{x}, y) , where $\mathbf{x} \in \mathbb{R}^d$ is a feature vector in d -dimensional space, and $y \in \{\mathcal{Y}_1, \dots, \mathcal{Y}_{\kappa}\}$ (κ denotes the number of classes) is the *label* (for the training set) or the *prediction* (for the testing set). The process of classification is then to build a statistical model that can operate the input feature vector \mathbf{x} to the output as its class y . In this section, the classification models involved in this thesis will be introduced.

2.4.1 Classical Models

A series of classical models in the machine learning community are introduced in this section. Generally, these models have a simplified mathematical theory and can be efficient for some tasks within a limited data size. As general audio signal classification is a young field with a small size of labelled data, specifically, healthcare related audio data (e.g., snore sounds) are extremely expensive for human expert annotation, the classical models are investigated as the fundamental research basis.

Naïve Bayes

The *Naïve Bayes* (NB) classifier is based on a conditional probability model, which assigns the given instance's probability as [143]:

$$P(y|\mathbf{x}) = P(y|x_1, \dots, x_d) = \frac{P(y)P(x_1, \dots, x_d|y)}{P(x_1, \dots, x_d)}, \quad (2.20)$$

where (x_1, \dots, x_d) is the feature vector. For Naïve Bayes classifiers, the assumption is made that each feature is independent of the value of the other features when given the *class* variable. Therefore, Equation 2.20 can be simplified as:

$$P(y|\mathbf{x}) = \frac{P(y) \prod_{m=1}^d P(x_m|y)}{P(x_1, \dots, x_d)}, \quad (2.21)$$

where m denotes the index of the feature value in the vector. As $P(x_1, \dots, x_d)$ will be a constant when the feature values x_1, \dots, x_d are known, there will be a relationship inferred from Equation 2.21 as:

$$P(y|\mathbf{x}) \propto P(y) \prod_{m=1}^d P(x_m|y). \quad (2.22)$$

When constructing a classifier, the *maximum a posteriori* (MAP) [144] decision rule is used:

$$\hat{y} = \arg \max_y P(y) \prod_{m=1}^d P(x_m|y), \quad (2.23)$$

where \hat{y} is the *prediction*, and $P(y)$ is the relative frequency of a class variable in the training set. To estimate the distribution $P(x_m|y)$, there are various parametric methods like *Gaussian distribution estimation* or nonparametric methods like *kernel density estimation* [145].

In spite of the over-simplified assumptions, the NB classifier can work very well in many fields, e.g., document classification [146] and spam filtering [147]. More details on a theoretical analysis of NB's strength can be found in [148].

k-Nearest-Neighbour

The *k*-nearest-neighbour (*k*-NN) classifier is a variant of the NN classifier (where $k = 1$) [143, 149], which searches the nearest neighbour \mathbf{x}_σ from *training* instances to a given *test* instance \mathbf{x}_t . The label of \mathbf{x}_σ , i.e., y_σ , will be assigned to \mathbf{x}_t as the *prediction*. For a *k*-NN classifier (when $k > 1$), it searches k nearest neighbours rather than only one nearest neighbour as the NN classifier. The majority class variable of these k nearest neighbours will be assigned to the given *test* instance as the *prediction* [143]. One common method to find the nearest neighbours is by measuring the *Euclidean* distance [150], which is defined as:

$$D(\mathbf{x}_t, \mathbf{x}_i) = D(\mathbf{x}_i, \mathbf{x}_t) = \sqrt{\sum_{m=1}^d (\mathbf{x}_{t,m} - \mathbf{x}_{i,m})^2}, \quad (2.24)$$

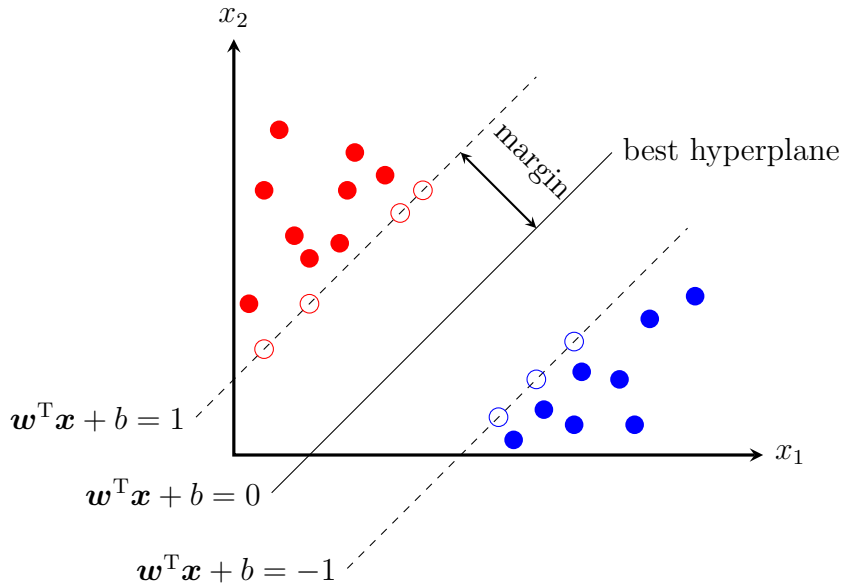


Figure 2.8: Mechanism of training an SVM classifier in a binary classification problem. The aim of training an SVM classifier is to find the best hyperplane which can be achieved by maximising the margin. Data points with the widest possible gap are called support vectors (indicated by circles); \mathbf{w} is a normal vector; b is a bias.

where $D(\mathbf{x}_t, \mathbf{x}_i)$ means the calculated distance between the given *test* instance \mathbf{x}_t and a certain instance \mathbf{x}_i in the *training* set. Previous studies had demonstrated the effectiveness of k -NN in classification of snore sounds [15–17]. In this thesis, k -NN will be investigated in comparison to other classifiers for SnS classification.

Support Vector Machines

Support vector machines (SVMs) [81] aim to find a set of hyperplanes in a multi-dimensional space such that instances of different *class* variables can be separated. More precisely, it is the goal of an SVM to find the best hyperplane that maximises the separation between classes. In other words, this hyperplane has the largest distance (also known as *margin*) to the nearest training data point of any class [89]. When performing classification tasks, a subset of data points with the widest possible gap (called as *support vectors*) from the *training* set will be selected as pivots to support the hyperplane on both sides of the *margin*. The instances from the *test* set will be mapped to this multi-dimensional space, and the *predictions* will be given based on which side of the gap they fall onto. Figure 2.8 briefly shows the mechanism of training an SVM classifier in a binary classification problem.

Formally, in a binary classification problem (e. g., $y_i \in \{-1, +1\}$), SVMs find the

optimal margin separating hyperplane by solving the optimisation problem [81]:

$$\begin{aligned}
 \text{minimise : } & \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^n \alpha_i, \\
 \text{subject to : } & \sum_{i=1}^n \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C_s, \quad i = 1, \dots, n,
 \end{aligned} \tag{2.25}$$

where α_i corresponds to the *Lagrange* multiplier of a training sample (\mathbf{x}_i, y_i) , and C_s is a pre-defined parameter. $K(\mathbf{x}_i, \mathbf{x}_j)$ is called *kernel function* [81], which can make SVMs analyse linearly or nonlinearly separable problems. There are some commonly used kernel functions, e. g., *linear*, *polynomial*, and *radial basis function* (RBF). The definitions of these kernel functions can be expressed as follows:

$$\text{linear : } K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j, \tag{2.26a}$$

$$\text{polynomial : } K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i^T \mathbf{x}_j + c)^{\hat{d}}, \tag{2.26b}$$

$$\text{RBF : } K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}, \tag{2.26c}$$

where γ , c , and \hat{d} are pre-defined parameters.

To solve the aforementioned optimisation problem, the *sequential minimal optimisation* (SMO) algorithm can be used [151]. For classification of a given *test* sample \mathbf{x}_t , a decision function is defined as:

$$f(\mathbf{x}_t) = \sum_{i=1}^n \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_t) + b, \tag{2.27}$$

where b is the *bias*.

There are two popular methods which can combine several binary SVMs to solve the multi-class problems: *One-versus-all* trains one binary SVM classifier for each class, and then, the prediction of a *test* sample will be given by which SVM classifier has the highest output function; *One-versus-one* trains one binary SVM classifier for each pair of classes, and then, the prediction of a *test* sample will be given by which class has the most votes [152]. In the past two decades, SVMs have been the popular classifiers for various applications in machine learning, which can achieve a satisfying recognition result.

Random Forests

Random forests (RFs) [153] are a combination of decision trees [154], which can efficiently ease the influence of overfitting by a single trained decision tree. As a kind of *ensemble learning* [155] algorithms, RFs use several individual trees to make the final prediction via the ‘bagging’ algorithm [156]. In the paradigm of ‘bagging’

(or bootstrap aggregating), *weak learners* (i. e., individual trees in RFs) are firstly trained with different sets of bootstrap examples, i. e., randomly selected subsampled training data sets with replacement (in the RFs algorithm, the subspace of the whole features will also be randomly selected). Then, the final prediction will be given by a *majority voting* (cf. Section 2.6) of the trained weak learners (i. e., individual trees in RFs). Compared with a single decision tree classifier, RF classifier can be more robust and generalised in real applications.

2.4.2 Deep Learning Models

Deep learning (DL) has now become a very popular subset of machine learning since the greedy layer-wise unsupervised pre-training was proposed to train very deep neural networks in 2006 [157, 158]. The core idea of DL models is to extract higher representations from the data with the help of a series of nonlinear transformation of the inputs. More precisely, it is expected to learn more robust and generalised features via DL models from a big data size, which was restrained by the capacity of the aforementioned classical models (usually with a shallow architecture). The success of DL has been proven in many fields like speech recognition [159], image recognition [160], or object detection [161], etc. A recent study on snore sound classification [93] demonstrated that, simple subband energy features via DL models can reach a better performance compared with more sophisticated features like wavelets. In this section, some fundamental concepts of the neural networks will be introduced firstly. Then, DL models involved in this thesis will be described. It should be noted that, *convolutional neural networks* (CNNs) [162] have been a popular DL model to serve as an efficient feature extractor for classification task. When using a CNN-based feature extractor, the audio signal can for example be transformed to images like spectrograms (via Fourier transformation) or scalograms (via wavelet transformation). This thesis focuses on using acoustic features derived from audio signals, which does not involve CNN models. For relevant work on using CNNs, the reader is referred to the works in [39, 41, 163–166].

Multilayer Perceptrons

Multilayer perceptrons (MLPs) belong to the feed-forward neural networks (FNNs), a kind of simple network architecture with connected *neurons* (nodes) feeding forward from one layer to the next one [167]. Typically, in MLP, each layer is fully connected with all nodes of the subsequent layer, however, there are no connections between nodes within the same layer or across multiple layers. Figure 2.9 gives an example of a two-hidden-layer MLP.

Formally, given the input vector $\mathbf{h}^{l-1} \in \mathbb{R}^m$ to the l -th layer, its output vector

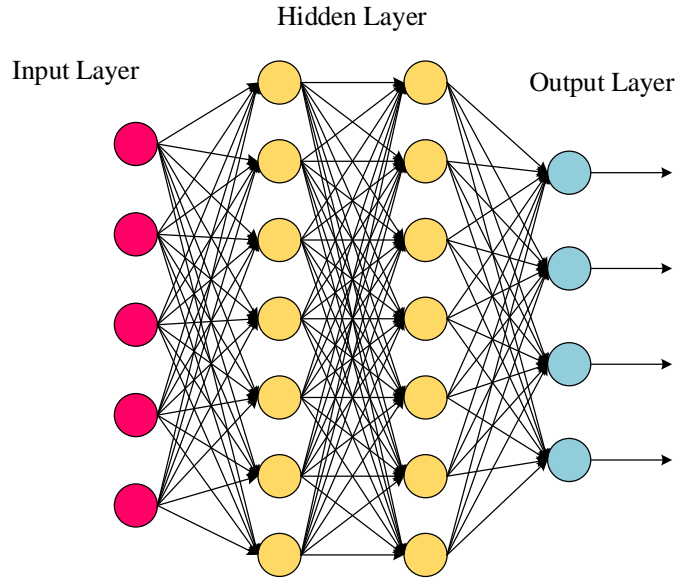


Figure 2.9: Example of a two-hidden-layer MLP.

$\mathbf{h}^l \in \mathbb{R}^n$ can be written as:

$$\mathbf{h}^l = \mathcal{F}(\mathbf{W}^l \mathbf{h}^{l-1} + \mathbf{b}^l), \quad (2.28)$$

where $\mathbf{W} \in \mathbb{R}^{n \times m}$ and $\mathbf{b} \in \mathbb{R}^n$ are the *weight matrix* and the *bias vector*, respectively. $\mathcal{F}(\cdot)$ denotes a nonlinear and differentiable function, which is called *activation function*. The commonly used activation functions (see Figure 2.10) include the *sigmoid* function (sigm), the *hyperbolic tangent* function (tanh), and the *rectified linear unit* function (ReLU). These functions are defined as follows:

$$\text{sigm}(x) = \frac{1}{1 + e^{-x}}, \quad (2.29a)$$

$$\text{tanh}(x) = \frac{e^{2x} - 1}{e^{2x} + 1}, \quad (2.29b)$$

$$\text{ReLU}(x) = \max(0, x). \quad (2.29c)$$

It can be easily found that, the hyperbolic tangent function is a rescaled sigmoid function, i. e., $\text{tanh}(x) = 2\text{sigm}(2x) - 1$. Apart from these functions, there are other options like the *soft plus* [168] and the *maxout* [169]. There are many options to choose the activation functions for all hidden layers. For the output layer, the choice of the activation function depends on the task for building the MLP. In this thesis, general audio signal classification is usually a multi-class classification problem.

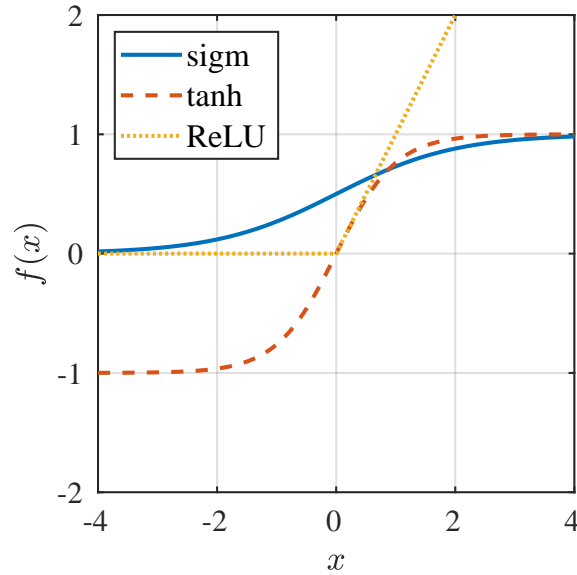


Figure 2.10: Some commonly used activation functions for neural networks: sigmoid (sigm), hyperbolic tangent (tanh), and rectified linear unit (ReLU).

Thus, the *softmax* function is used [167]:

$$y_i = \frac{e^{x_i}}{\sum_{j=1}^{\kappa} e^{x_j}}, \quad i = 1, \dots, \kappa. \quad (2.30)$$

In this equation, y_i denotes the output of the i -th element (x_i) in the vector input to the softmax function, and κ is the number of output nodes (i.e., the number of classes). The output vector \mathbf{y} can be a valid probability distribution for $0 \leq y_i \leq 1, \forall i$ and the $\sum_{i=1}^{\kappa} y_i = 1$. For a given *test* sample, its *prediction* will be given as the node of the output layer which has the maximum probability value.

The procedure of training a neural network is to iteratively update the parameters of its layers (\mathbf{W} and \mathbf{b}) in order to minimise the *loss function* $\mathcal{L}(\boldsymbol{\theta})$ ($\boldsymbol{\theta}$ denotes the parameters), which measures the difference between the target output vectors and the actual output vectors of the network. For an instance \mathbf{x} fed as the input to the network, \mathbf{y} as the actual output, \mathbf{t} as the target output, the *cross entropy* is usually adopted in multi-class classification task as the loss function:

$$\mathcal{L}(\boldsymbol{\theta}) = H(\mathbf{y}, \mathbf{t}) = - \sum_{i=1}^{\kappa} t_i \log y_i. \quad (2.31)$$

One popular method to minimise the loss function is called *backpropagation* (BP) [170], which repeatedly applies the chain rule of differentiation to compute the

gradient $\nabla\mathcal{L}(\boldsymbol{\theta})$ of the loss function with respect to the parameters of the network. The detailed formula derivation and implementation of BP can be found in [167, 170, 171]. The *gradient descent* algorithm is usually used to adjust the network parameters in small steps towards the direction of the negative gradient [170]:

$$\boldsymbol{\theta}_{\tau+1} = \boldsymbol{\theta}_{\tau} - \eta\nabla\mathcal{L}(\boldsymbol{\theta}_{\tau}), \quad (2.32)$$

where τ denotes the iteration step and $\eta > 0$ is the *learning rate* which should be allowed to vary in the learning process [172]. The *stochastic gradient descent* (SGD) [173] is often used in order to accelerate the process of updating the network parameters when computing the gradient on the entire training set. In SGD, the update to the network parameters will be based on the gradient value of the loss function for one instance only. In practice, the network parameters will be repeatedly updated by applying SGD to *minibatches* which include a set of instances divided from the entire train set. The process of going over the entire training set is called an *epoch*. Usually the number of epochs will be set empirically dependent on different data sets and tasks.

Another technique to speed up the convergence of training neural network is to use the *momentum* [174]:

$$\boldsymbol{\theta}_{\tau+1} = \boldsymbol{\theta}_{\tau} - \eta\nabla\mathcal{L}(\boldsymbol{\theta}_{\tau}) + \mu(\boldsymbol{\theta}_{\tau} - \boldsymbol{\theta}_{\tau-1}), \quad (2.33)$$

where $\mu \in (0, 1)$ is the momentum term. An improvement of the momentum method is the *Nesterov momentum* [175]:

$$\boldsymbol{\theta}_{\tau+1} = \boldsymbol{\theta}_{\tau} - \eta\nabla\mathcal{L}(\boldsymbol{\theta}_{\tau} + \mu(\boldsymbol{\theta}_{\tau} - \boldsymbol{\theta}_{\tau-1})) + \mu(\boldsymbol{\theta}_{\tau} - \boldsymbol{\theta}_{\tau-1}). \quad (2.34)$$

Both momentum and Nesterov momentum can be regarded as indirect methods to change the learning rate to make a persistent reduction of the loss function across iterations.

Apart from the basic BP algorithm mentioned above, there are more sophisticated methods, e. g., the *scaled conjugate gradient* (SCG) [176], which avoids line-search in learning iterations, can be more efficient and faster than standard gradient descent. In addition, to cope with the overfitting problem, *regularisation* [177], *dropout* [178], and *batch normalisation* [179] are usually used.

Stacked Autoencoders

As one efficient kind of deep learning frameworks, *stacked autoencoders* (SAEs) can facilitate the network to learn higher representations from the inputs via an unsupervised fashion in the initialisation phase of the network [158, 180].

An *autoencoder* (AE) is a feed-forward neural network (often with one hidden layer) having an equal number of nodes between the inputs and the outputs [181,

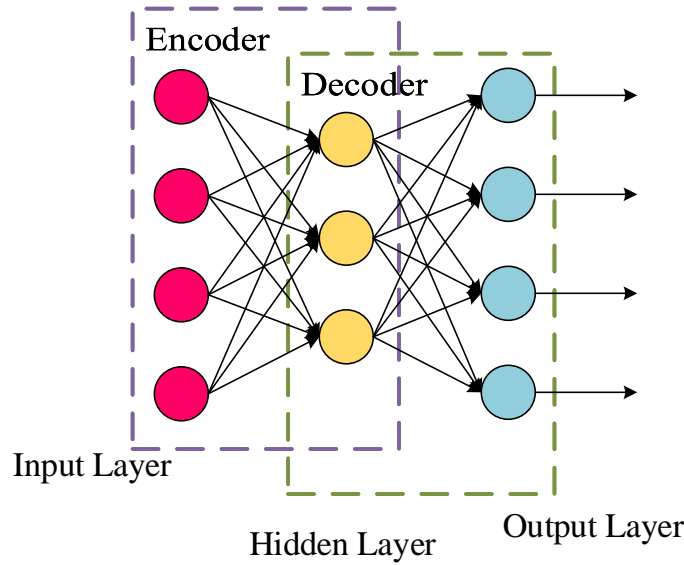


Figure 2.11: Structure of an autoencoder.

182]. Figure 2.11 shows an example of the structure of an autoencoder. Unlike the aforementioned MLP, training an AE is to reconstruct its inputs rather than give the predictions in the output layer. Typically, an AE is composed of two parts, i. e., an *encoder* and a *decoder*. Firstly, in the encoder stage of training an AE, the input vector $\mathbf{x} \in \mathbb{R}^m$ will be mapped onto a new representation $\mathbf{a} \in \mathbb{R}^n$ in the hidden layer as:

$$\mathbf{a} = \mathcal{F}(\mathbf{W}\mathbf{x} + \mathbf{b}), \quad (2.35)$$

where $\mathbf{W} \in \mathbb{R}^{n \times m}$ and $\mathbf{b} \in \mathbb{R}^n$ are the weight matrix and the bias vector, respectively. \mathcal{F} is the activation function as mentioned above. Then, in the decoder stage, the new representation \mathbf{a} will be mapped to the reconstruction $\mathbf{x}' \in \mathbb{R}^m$ as:

$$\mathbf{x}' = \mathcal{F}'(\mathbf{W}'\mathbf{a} + \mathbf{b}'), \quad (2.36)$$

where the weight matrix $\mathbf{W}' \in \mathbb{R}^{m \times n}$, the bias vector $\mathbf{b}' \in \mathbb{R}^m$, and the activation function \mathcal{F}' are the parameters that may differ in general from the corresponding \mathbf{W} , \mathbf{b} , and \mathcal{F} used in the encoder stage. The procedure of training an AE is to update the parameters $\boldsymbol{\theta} = \mathbf{W}, \mathbf{W}', \mathbf{b}, \mathbf{b}'$ by minimising the loss function $\mathcal{L}(\boldsymbol{\theta})$, e. g., *mean squared error* (MSE), as:

$$\mathcal{L}(\boldsymbol{\theta}) = E(\mathbf{x}, \mathbf{x}') = \frac{1}{N} \sum_{j=1}^N \|\mathbf{x}_j - \mathbf{x}'_j\|^2, \quad (2.37)$$

where \mathbf{x}_j represents the j -th input vector. The aforementioned optimisation methods like BP, SGD, or SCG can be applied to the AE training process. In addition,

to make the learnt representations from an AE more robust and generalised, *sparse coding* [183] and *regularisation* [167] are usually exploited to calculate the reconstruction error:

$$\mathcal{L}(\boldsymbol{\theta}) = E(\mathbf{x}, \mathbf{x}') = \frac{1}{N} \sum_{j=1}^N \|\mathbf{x}_j - \mathbf{x}'_j\|^2 + \alpha L_2 + \beta \sum_{k=1}^n SP(\rho || \hat{\rho}_k), \quad (2.38a)$$

$$L_2 = \frac{1}{2} \sum_{j=1}^N (\|\mathbf{W}_j\|^2 + \|\mathbf{W}'_j\|^2), \quad (2.38b)$$

$$SP(\rho || \hat{\rho}_k) = \rho \log \frac{\rho}{\hat{\rho}_k} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_k}, \quad (2.38c)$$

$$\hat{\rho}_k = \frac{1}{N} \sum_{j=1}^N h_k(\mathbf{x}_j), \quad (2.38d)$$

where L_2 is the L_2 regularisation term, $SP(\rho || \hat{\rho}_k)$ is the sparsity regularisation term [184] (which can be referred to as *Kullback-Leibler divergence* [185]), $\hat{\rho}_k$ is the average activation value of the k -th node, $\rho \in [0, 1)$ is the sparsity level, n is the number of hidden nodes, α and β are parameters for the L_2 regularisation term, and the sparsity regularisation term, respectively. Another variant of the basic AE, the *denoising autoencoder* (DAE) [186], is to reconstruct the inputs from a corrupted version, which can learn more robust features than the basic AE.

When constructing an SAE classifier, the learnt representation \mathbf{a}^l by the l -th encoder will be used as the inputs (features) to the subsequent $(l + 1)$ -th encoder. Firstly, the stacked autoencoders will be trained layer by layer via an unsupervised learning process to fulfil the pre-training phase for building the deep neural network. Then, a softmax layer will be often added as the output of the SAE architecture in a supervised learning process as when training the MLP described above.

Recurrent Neural Networks

When learning a sequence input (e.g., an audio clip includes some context information), the aforementioned FNNs will process each frame independently, which means using no context information. One simple and straightforward way is to stack several successive frames together as the input to the network [187]. However, this method can learn limited context information [188]. *Recurrent neural networks* (RNNs) [189] can learn context information by incorporating the outputs of a previous time step as additional inputs for the current time step (see Figure 2.12). In a RNN (referred to *Elman network* [189]), the output \mathbf{h}_t^l of the l -th hidden layer at the time t can be expressed as a modification from Equation 2.28:

$$\mathbf{h}_t^l = \mathcal{F}(\mathbf{U}^l \mathbf{h}_{t-1}^l + \mathbf{W}^l \mathbf{h}_t^{l-1} + \mathbf{b}^l), \quad (2.39)$$

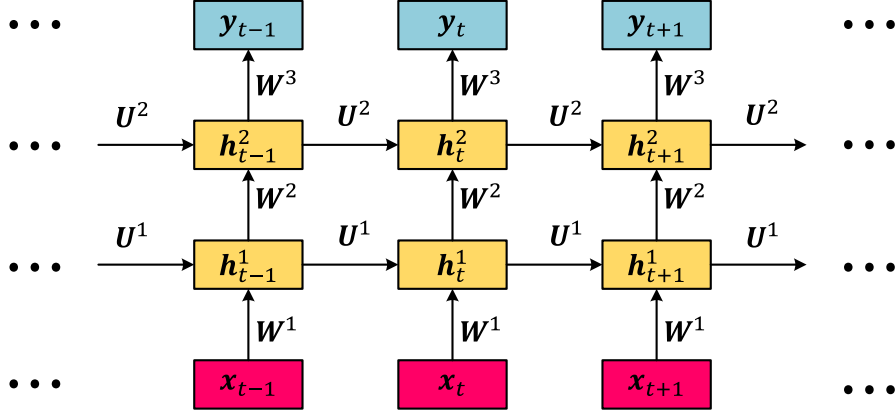


Figure 2.12: Structure of a RNN.

where \mathbf{U}^l is called *recurrent weight matrix* of the l -th hidden layer, \mathbf{W}^l and \mathbf{b}^l are the weight matrix and bias vector of the l -th hidden layer, \mathcal{F} denotes a nonlinear and differentiable function as mentioned in training an MLP.

RNNs can be trained via *backpropagation through time* (BPTT) [171], which is similar to BP for training MLP by repeatedly using the chain rule of differentiation. However, training the standard RNNs is difficult. The back-propagated error (repeatedly multiplied by the recurrent weight matrix) will be blown up (*gradient explosion*) or vanish (*vanishing gradient*) over time [190]. The gradient explosion will cause the training diverge, but it can be solved by *gradient clipping* [191], which clips the gradients into a limited range to prevent them from getting too large. The vanishing gradient problem will restrain an RNN to learn long-term context information. In practice, more complicated functions which have a *memory cell* to preserve long-term information are used to replace the traditional neurons in RNNs. There are two popular structures that can fulfil this work, i. e., *long short-term memory* (LSTM) cells [192] and *gated recurrent units* (GRUs) [87]. In this thesis, the RNNs with GRUs are exploited for their simplicity in structure and effectiveness in previous relevant studies [113, 165, 166]. The GRU [87] contains an *update gate* \mathbf{z} , a *reset gate* \mathbf{r} , an activation \mathbf{h} , and a candidate activation $\tilde{\mathbf{h}}$. Figure 2.13 illustrates the structure of a GRU. The mechanism of GRU can be expressed by the following equations:

$$\mathbf{r}_t^l = \mathcal{F}_\sigma(\mathbf{U}_r^l \mathbf{h}_{t-1}^l + \mathbf{W}_r^l \mathbf{h}_t^{l-1} + \mathbf{b}_r^l), \quad (2.40a)$$

$$\tilde{\mathbf{h}}_t^l = \mathcal{F}_h(\mathbf{U}_h^l (\mathbf{r}_t^l \odot \mathbf{h}_{t-1}^l) + \mathbf{W}_h^l \mathbf{h}_t^{l-1} + \mathbf{b}_h^l), \quad (2.40b)$$

$$\mathbf{z}_t^l = \mathcal{F}_\sigma(\mathbf{U}_z^l \mathbf{h}_{t-1}^l + \mathbf{W}_z^l \mathbf{h}_t^{l-1} + \mathbf{b}_z^l), \quad (2.40c)$$

$$\mathbf{h}_t^l = \mathbf{z}_t^l \odot \tilde{\mathbf{h}}_t^l + (1 - \mathbf{z}_t^l) \odot \mathbf{h}_{t-1}^l, \quad (2.40d)$$

where \odot denotes the element-wise multiplication, \mathcal{F}_σ is the sigmoid function (sigm)

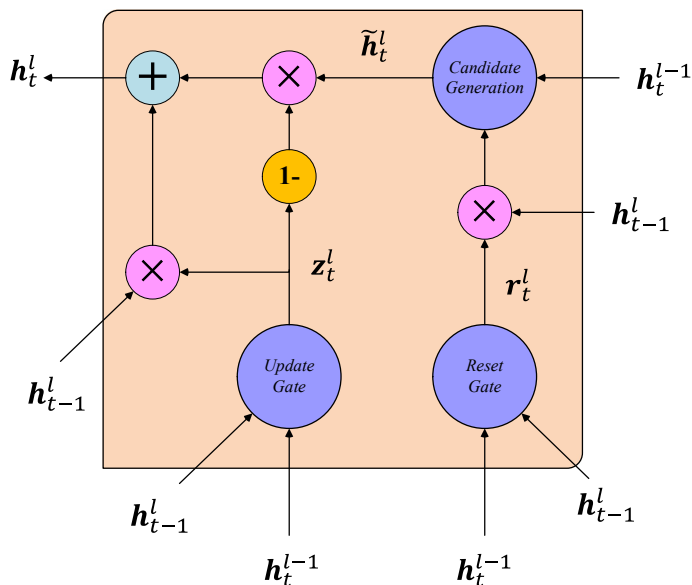


Figure 2.13: Structure of a Gated Recurrent Unit (GRU).

to scale values to the range between 0 and 1, and \mathcal{F}_h is usually set to the hyperbolic tangent function (tanh). As indicated in [87], the unit will not be overwritten if the update gate is closed (gate activation values are close to 0), which helps to remember the existing context information from inputs for a long series of time steps. In addition, the error can be back-propagated without too much attenuation by passing through the update gate when it is open (gate activation values are close to 1), which solves the vanishing gradient problem in standard RNNs. In this thesis, the RNNs using GRUs are referred to as *Gated Recurrent Neural Networks* (GRNNs) for conciseness.

Bidirectional Recurrent Neural Networks

The aforementioned RNNs only consider the information flow in one direction, which ignores the future context information. Another structure (see Figure 2.14), *bidirectional recurrent neural networks* (BRNNs) [88] can access the whole context information of inputs by calculating the forward hidden layer activation $\vec{\mathbf{h}}$ (*forward chain*: from the beginning to the end of the sequence), and the backward hidden layer activation $\overleftarrow{\mathbf{h}}$ (*backward chain*: from the end to the beginning of the sequence):

$$\vec{\mathbf{h}}_t^l = \mathcal{F}(\vec{\mathbf{U}}^l \vec{\mathbf{h}}_{t-1}^l + \mathbf{W}^l \mathbf{h}_t^{l-1} + \vec{\mathbf{b}}^l), \quad (2.41a)$$

$$\overleftarrow{\mathbf{h}}_t^l = \mathcal{F}(\overleftarrow{\mathbf{U}}^l \overleftarrow{\mathbf{h}}_{t+1}^l + \mathbf{W}^l \mathbf{h}_t^{l-1} + \overleftarrow{\mathbf{b}}^l), \quad (2.41b)$$

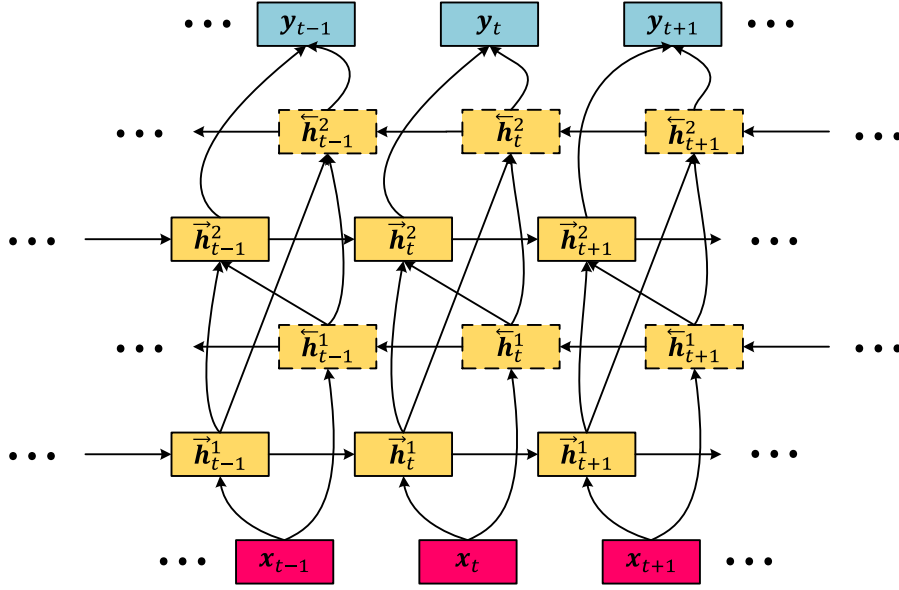


Figure 2.14: Structure of a bidirectional RNN.

where \vec{U}^l , \vec{b}^l are the forward recurrent weight matrix and bias vector, and \overleftarrow{U}^l , \overleftarrow{b}^l are the backward recurrent weight matrix and bias vector. The activation \mathbf{h}_t^l will be the concatenation of the two chains. Correspondingly, in this thesis, the BRNNs using GRUs are referred to as *Bidirectional Gated Recurrent Neural Networks* (BGRNNs).

2.4.3 Extreme Learning Models

In the past decade, the extreme learning machines (ELMs) [82, 193, 194] and its variants (a review is found in [195]) have been popular in the community of neural network and machine learning. It is reported that, the extreme learning models (ELM and its variants) have achieved superior performance to conventional models like SVMs and MLPs in many applications of regression and classification [195]. The universal approximation and classification capabilities of ELM have been proven in theory [196–198].

Extreme Learning Machines

Extreme learning machines (ELMs) [193] are one kind of single hidden layer feed-forward neural networks (SLFNNs). Figure 2.15 shows the structure of an SLFNN. Unlike the conventional SLFNNs trained with gradient-based methods, there is no need for tuning the parameters of the hidden nodes for ELMs [199, 200]. For ELMs, the hidden nodes are randomly initiated and the output weights can be analytically

determined [194]. More specifically, the parameters of ELMs are independent of the training data, i.e., the parameters of ELMs can be generated before seeing the training data [199]. Formally, the output function of an ELM for generalised SLFNNs (one output node as an example) can be written as [82, 199]:

$$f(\mathbf{x}) = \mathbf{h}(\mathbf{x})\mathbf{w} = \sum_{l=1}^L w_l h_l(\mathbf{x}), \quad (2.42)$$

where $\mathbf{x} \in \mathbb{R}^D$ is the input vector, w_l is the output weight between the l -th hidden node and the output node, and $h_l(\mathbf{x}) = \mathcal{G}(\mathbf{a}_l, b_l, \mathbf{x})$ is the output of the l -th hidden node. $\mathbf{h}(\mathbf{x})$ maps the input \mathbf{x} from a D -dimensional space to an L -dimensional space (*ELM feature space*). The parameters $\mathbf{a}_l \in \mathbb{R}^D$ and b_l are input weights and bias in the l -th hidden node, which are assigned randomly in the ELM training process. The activation function $\mathcal{G}(\cdot)$ can be a nonlinear piecewise continuous function which satisfies the ELM universal approximation capability theorems [196–198]. Except for the sigmoid function (see Section 2.4.2), there are some other commonly used activation functions (see Figure 2.16) for ELM, like the *Fourier* function (Fr), the *hard-limit* function (hardlim), the *triangle basis* function (tribas), or the *radical basis* function (radbas), which are defined as follows:

$$\text{Fr}(x) = \sin x; \quad (2.43a)$$

$$\text{hardlim}(x) = \begin{cases} 1, & \text{if } x \geq 0, \\ 0, & \text{otherwise;} \end{cases} \quad (2.43b)$$

$$\text{tribas}(x) = \begin{cases} 1, & \text{if } x \in [-1, 1], \\ 0, & \text{otherwise;} \end{cases} \quad (2.43c)$$

$$\text{radbas}(x) = e^{-x^2}. \quad (2.43d)$$

Given a set of training examples $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]$ with the target matrix $\mathbf{T} \in \mathbb{R}^{N \times M}$, ELM aims to minimise not only the training error $\|\mathbf{H}\mathbf{W} - \mathbf{T}\|^2$, but also the norm of the output weights $\|\mathbf{W}\|$ [82]. $\mathbf{H} \in \mathbb{R}^{N \times L}$ is the output matrix of the hidden layer:

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}_1) & \cdots & h_L(\mathbf{x}_1) \\ \vdots & \vdots & \vdots \\ h_1(\mathbf{x}_N) & \cdots & h_L(\mathbf{x}_N) \end{bmatrix}, \quad (2.44)$$

$\mathbf{W} \in \mathbb{R}^{L \times M}$ is the output weight matrix, and $\mathbf{T} \in \mathbb{R}^{N \times M}$ is the target matrix. M denotes the number of output nodes, i.e., the number of classes in a classification problem. In the original implementations of ELMs [193, 194], the minimal norm least square method was used to calculate the output weights as:

$$\mathbf{W} = \mathbf{H}^\dagger \mathbf{T}, \quad (2.45)$$

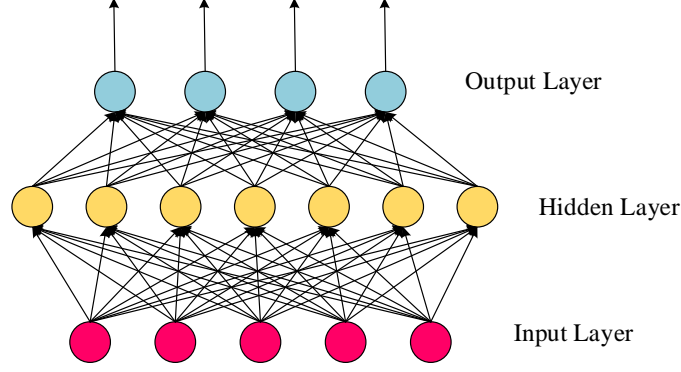


Figure 2.15: Structure of an SLFNN.

where $\mathbf{H}^\dagger \in \mathbb{R}^{L \times N}$ is the *Moore-Penrose generalised inverse* of the matrix \mathbf{H} [201, 202]. As suggested in [82], two methods are usually used to accelerate the calculation of the output weights as:

$$\mathbf{W} = \mathbf{H}^\top \left(\frac{\mathbf{I}}{C_e} + \mathbf{H}\mathbf{H}^\top \right)^{-1} \mathbf{T}, \quad \mathbf{I} \in \mathbb{R}^{N \times N}, \quad (2.46)$$

$$\mathbf{W} = \left(\frac{\mathbf{I}}{C_e} + \mathbf{H}^\top \mathbf{H} \right)^{-1} \mathbf{H}^\top \mathbf{T}, \quad \mathbf{I} \in \mathbb{R}^{L \times L}, \quad (2.47)$$

where \mathbf{I} is an identity matrix, and C_e is a pre-defined parameter. As indicated by Huang *et al.* in [82], one may use Equation 2.46 when the number of training instances is not huge ($N \ll L$), or apply Equation 2.47 when the number of training instances is huge ($L \ll N$), in order to reduce the computational costs. In this thesis, to fully investigate the capacity of ELMs, the number of hidden nodes will be tuned and tested including large values (more than 5 000). Thus, Equation 2.46 is used, by which the output function of an ELM classifier can be expressed as:

$$\mathbf{f}(\mathbf{x}) = \mathbf{h}(\mathbf{x})\mathbf{W} = \mathbf{h}(\mathbf{x})\mathbf{H}^\top \left(\frac{\mathbf{I}}{C_e} + \mathbf{H}\mathbf{H}^\top \right)^{-1} \mathbf{T}, \quad (2.48)$$

where $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_M(\mathbf{x})]^\top$ is the output vector. In a multi-class problem, the label (prediction) y of an instance \mathbf{x} is given as the index of the output node which has the highest output value with respect to the input instance:

$$y = \underset{m \in \{1, \dots, M\}}{\operatorname{argmax}} f_m(\mathbf{x}). \quad (2.49)$$

Kernel-based Extreme Learning Machines

If the feature mapping $\mathbf{h}(\mathbf{x})$ is unknown, the kernel trick $K(\mathbf{x}_i, \mathbf{x}_j)$ (similar to SVM, see Equation 2.26) can be used. In this case, Equation 2.46 will be used, and the

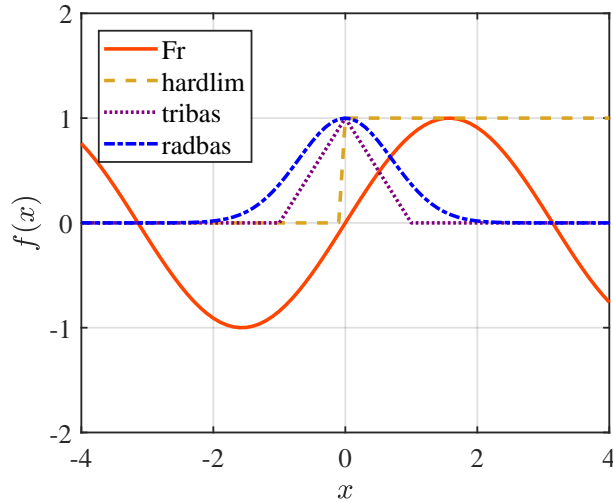


Figure 2.16: Some commonly used activation functions for extreme learning machines: Fourier (Fr), hard-limit (hardlim), triangle basis (tribas), and radical basis (radbas).

output function can be expressed as [82]:

$$\begin{aligned}
 \mathbf{f}(\mathbf{x}) &= \mathbf{h}(\mathbf{x})\mathbf{W} = \mathbf{h}(\mathbf{x})\mathbf{H}^T \left(\frac{\mathbf{I}}{C_e} + \mathbf{H}\mathbf{H}^T \right)^{-1} \mathbf{T} \\
 &= \begin{bmatrix} K(\mathbf{x}, \mathbf{x}_1) \\ \vdots \\ K(\mathbf{x}, \mathbf{x}_N) \end{bmatrix} \left(\frac{\mathbf{I}}{C_e} + \boldsymbol{\Omega}_{ELM} \right)^{-1} \mathbf{T}, \tag{2.50}
 \end{aligned}$$

where $\boldsymbol{\Omega}_{ELM} = \mathbf{H}\mathbf{H}^T$ is the kernel matrix. In this thesis, to make the above expressed method distinct with the aforementioned ELMs (with random feature mappings), the name *kernel-based extreme learning machines* (KELMs) is used. Compared with KELMs, SVMs may tend to reach sub-optimal solutions when the same kernels are used [199].

2.5 Data Enrichment

In general, *data enrichment* is often needed when integrating the available data in real world [89]. In this thesis, the main focus is to use *active learning* (AL) for efficient labelling. Another typical approach for data enrichment is called *semi-supervised learning* (SSL) [203], which can exploit the annotation work of unlabelled data in a non-human involved scenario. However, for general audio data (e. g., bird sound data), some domain knowledge from the human experts (e. g., ornithologists)

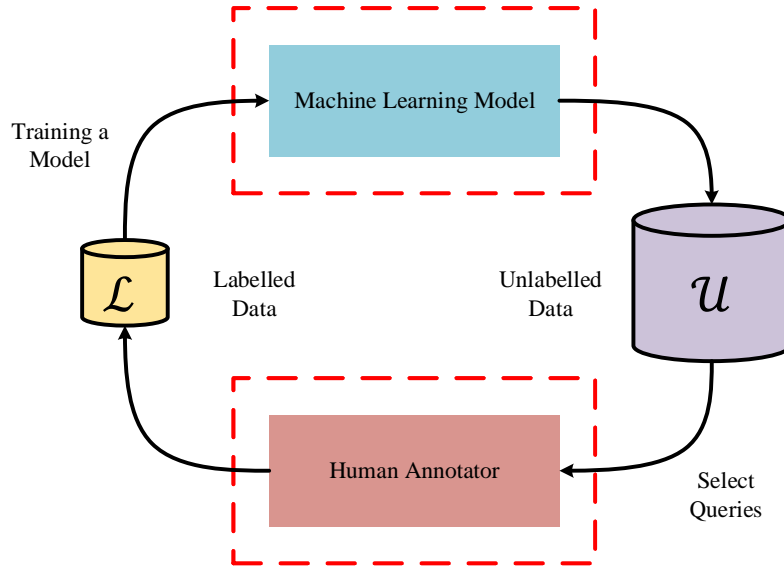


Figure 2.17: Pool-based active learning scenario.

Algorithm 3: Passive Learning (PL)

- 1 **repeat**
 - 2 Randomly select K samples \mathcal{D}_K from the pool of unlabelled data \mathcal{U} .
 - 3 Let human expert annotate the selected subset \mathcal{D}_K .
 - 4 Remove \mathcal{D}_K from the unlabelled data \mathcal{U} : $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{D}_K$.
 - 5 Add \mathcal{D}_K to the labelled data \mathcal{L} : $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{D}_K$.
 - 6 **until** *iteration reaches a pre-defined number, or the trained model achieves a certain performance on the validation set.*
-

could still be important to guarantee efficient annotations. Thus, AL was chosen to be investigated in this thesis. There are several different scenarios of AL like *membership query synthesis*, *stream-based selective sampling*, or *pool-based sampling* [204]. In this thesis, AL is applied to a pool-based sampling scenario (see Figure 2.17), in which, there is a small set of labelled data \mathcal{L} and a large pool of unlabelled data \mathcal{U} [204].

2.5.1 Passive Learning

In contrast to active learning, *passive learning* (PL) does not consider involving the previously trained model to participate. PL randomly selects the unlabelled data to query an oracle (e. g., a human expert) for annotation [204]. As indicated in [83], this method is extremely time-consuming and costly. The detailed steps of PL are

Algorithm 4: Sparse-Instance-based Active Learning (SI-AL)

```

1 repeat
2   Train a model  $\mathcal{C}$  based on the labelled data  $\mathcal{L}$ .
3   Randomly select  $K$  samples  $\mathcal{D}_k$  from the pool of unlabelled data  $\mathcal{U}$  that
   are predicted by  $\mathcal{C}$  as belonging to the sparse class.
4   Let human expert annotate  $\mathcal{D}_K$ .
5   Remove  $\mathcal{D}_K$  from the unlabelled data  $\mathcal{U}$ :  $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{D}_K$ .
6   Add  $\mathcal{D}_K$  to the labelled set  $\mathcal{L}$ :  $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{D}_K$ .
7 until iteration reaches a pre-defined number, or the trained model  $\mathcal{C}$  achieves
   a certain performance on the validation set.

```

briefly shown in Algorithm 3.

2.5.2 Active Learning

AL uses a *query strategy* based on the evaluation of a previously trained model to ask for human annotation. By selecting the ‘most informative’ unlabelled data from the pool, AL aims to improve the model’s performance using as few human annotated instances as possible [204]. There are a variety of AL query strategies that can be chosen for evaluating the informativeness of unlabelled data (details can be found in [204]). In this thesis, considering simplicity and effectiveness in real-world applications (e. g., for bird sound data), two strategies are investigated and compared, i. e., sparse-instance-based AL (SI-AL), and least-confidence-score-based AL (LCS-AL).

Sparse-Instance-based Active Learning

In SI-AL (see Algorithm 4), the sparsity of certain classes in the unbalanced data is taken into account. The instances which are predicted by the previously trained model to a certain *sparse* class will be selected for asking for human annotation. Formally, the query function is defined as:

$$Q_{SI}(\mathbf{x}) = \begin{cases} 1, & \text{if } \hat{y}_{\mathbf{x}} \in \mathcal{Y}_{sparse}, \\ 0, & \text{otherwise,} \end{cases} \quad (2.51)$$

where $\hat{y}_{\mathbf{x}}$ is the predicted label of instance \mathbf{x} , \mathcal{Y}_{sparse} is a set of sparse classes. Intuitively, \mathcal{Y}_{sparse} can be pre-defined in a global view of the whole data, or dynamically re-defined through the AL iterative process. In a real world application, some certain sparse classes are usually known to the data mining developers (e. g., bird sound data). Besides, dynamically changing the sparse classes may cause an unstable performance of the model. Therefore, in this thesis, \mathcal{Y}_{sparse} is pre-defined before the AL

Algorithm 5: Least-Confidence-Score-based Active Learning (LCS-AL)

-
- 1 **repeat**
 - 2 Train a classifier \mathcal{C} based on the labelled data \mathcal{L} .
 - 3 Predict the unlabelled data \mathcal{U} by \mathcal{C} , and rank the data by its prediction *confidence score*.
 - 4 Randomly select K samples \mathcal{D}_K from the last $\lambda_c N_{\mathcal{U}}$ of the ranked data in \mathcal{U} , $N_{\mathcal{U}}$ is the number of instances in \mathcal{U} , λ_c is a pre-defined factor.
 - 5 Let human expert annotate \mathcal{D}_K .
 - 6 Remove \mathcal{D}_K from the unlabelled data \mathcal{U} : $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{D}_K$.
 - 7 Add \mathcal{D}_K to the labelled data \mathcal{L} : $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{D}_K$.
 - 8 **until** *iteration reaches a pre-defined number, or the trained model \mathcal{C} achieves a certain performance on the validation set.*
-

process. SI-AL was previously proposed in a binary case [83]. In that case, \mathcal{Y}_{sparse} can simply be assigned to one of the two classes. In this thesis (a multi-class case), a *sparseness factor* λ_s is introduced to define \mathcal{Y}_{sparse} as:

$$\mathcal{Y}_{sparse} = \bigcup_{i=1}^{N_s} \{\mathcal{Y}_i | N_{y_i} < \lambda_s N_{max}\}, \quad (2.52)$$

where N_s is the total number of sparse classes, N_{y_i} is the number of instances that belong to the class \mathcal{Y}_i , and N_{max} is the number of instances that belong to one certain class which occupies the biggest proportion in the whole data. It should be noted that SI-AL will temporarily perform a PL paradigm if there are no instances assigned to the sparse classes.

Least-Confidence-Score-based Active Learning

In LCS-AL (see Algorithm 5), the query framework is based on *uncertainty sampling* [204, 205], in which the unlabelled instances with least uncertainty for a previously trained model to label will be selected for asking for human annotation. Formally, the query function can be defined as:

$$Q_{LCS}(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} = \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmin}} CS(\mathbf{x}), \\ 0, & \text{otherwise,} \end{cases} \quad (2.53)$$

where $CS(\cdot)$ denotes a function to calculate the *confidence score* of a previously trained model when it is making the prediction for instance \mathbf{x} . Some commonly used methods to calculate include *least confident*, *margin sampling*, and *entropy* [204]. However, the best method may be application-dependent due to each method having

its own strengths and weaknesses [204]. In this thesis, the margin sampling [206] is used for its ability to reduce the classification error by bettering the model on discriminating among specific classes [204]. The CS value based on margin sampling can be defined as [204]:

$$CS(\mathbf{x}) = P_{\mathcal{C}}(\hat{y}_1 | \mathbf{x}) - P_{\mathcal{C}}(\hat{y}_2 | \mathbf{x}), \quad (2.54)$$

where \hat{y}_1 and \hat{y}_2 are the predicted labels corresponding to the first and second highest posterior probability under the trained model \mathcal{C} . It can be understandable that instances with smaller margins will be more ambiguous to the model than those with larger margins. Thus, knowing the true labels of such instances can contribute to the improvement of the model's performance [204]. To estimate the posterior probability, the methods can be varied based on different classification models. For neural networks (e. g., DNNs, ELMs, or RNNs), the posterior probabilities can be the outputs passing a softmax function. For SVMs (involved in this thesis), the posterior probability can be estimated by using a logistic regression of the SVM's outputs (in the binary case) [207]. In a multi-class case, the aforementioned method can be extended to [208] (implemented in the WEKA toolkit [209]) or [210] (implemented in the LIBSVM toolkit [211]).

2.6 Late Fusion

It is expected to better the system's performance by an *early fusion* (feature fusion) and, or *late fusion* (model fusion). In a previous study [21], early fusion involved feature selection and reduction, which might be much dependent on human experience, time-consuming, and unstable. In this thesis, two simple and efficient late fusion strategies are employed, i. e., *majority voting* (MV) and *margin sampling voting* (MSV). For both the two strategies, models (classifiers) are trained dependently at first. Then, the trained models are combined together as members of a *committee*. The final prediction will be given by evaluating each member's prediction with a certain strategy.

Majority Voting

In the scenario of majority voting, the final prediction will be assigned to a class which appears mostly among all the predictions of the independently trained models. Let $d_{i,j} \in \{0, 1\}$ denote the decision value of the i -th model for the j -th class, the behaviour of MV can be described as:

$$\tilde{y} = \{\mathcal{Y}_j | \sum_{i=1}^{\mathcal{N}} d_{i,\mathcal{Y}_j} = \max_{j=1}^{N_c} \sum_{i=1}^{\mathcal{N}} d_{i,j}\}, \quad (2.55a)$$

$$d_{i,j} = \begin{cases} 1, & \mathcal{Y}_i = \mathcal{Y}_j, \\ 0, & \text{otherwise,} \end{cases} \quad (2.55b)$$

where \mathcal{N} is the number of models, N_c is the number of classes, \mathcal{Y}_i is the prediction of i -th model, \mathcal{Y}_j is the label of j -th class, and \tilde{y} is the final prediction. When the predictions of the models are not consistent with each other, or there is more than one class that has the most votes, the final prediction will be given as:

$$\tilde{y} = \{\mathcal{Y}_i | \operatorname{argmax}_{i=1}^{\mathcal{N}} \mathcal{W}_{i,\mathcal{Y}_i}\}, \quad (2.56)$$

where $\mathcal{W}_{i,\mathcal{Y}_i}$ is a calibration weight for the i -th model to make its prediction as \mathcal{Y}_i . In this thesis, $\mathcal{W}_{i,\mathcal{Y}_i}$ is adopted as the *Recall* (see Equation 2.58) of the i -th model for class \mathcal{Y}_i .

Margin Sampling Voting

In the scenario of margin sampling voting, the final prediction will be assigned to the class of a model which has the highest margin sampling value (see Equation 2.54) in the committee. The behaviour of MSV can be described as:

$$\tilde{y} = \{\mathcal{Y}_i | \operatorname{argmax}_{i=1}^{\mathcal{N}} \mathcal{M}_i\}, \quad (2.57)$$

where \mathcal{M}_i is the margin sampling value of i -th model. Therefore, this strategy is based on measuring each model's confidence when making the decision. The same as for MV, the calibration weight (see Equation 2.56) will be given once there are more than one model that can have the maximum margin sampling value.

2.7 Evaluation Metrics

In this section, the evaluation metrics for measuring the classification performance and the significance level will be given a brief description. Evaluations are usually done for measuring the performance of a trained system. Therefore, the following relevant instance numbers are referred to as the numbers in the development or the test sets.

2.7.1 Classification Evaluation

In a classification task, the evaluations are usually based on comparing the predicted labels and the ground truth. Recall (or *class-wise accuracy*) is the proportion of the instances that are correctly predicted among all the instances that belong to one certain class. In a multi-class classification problem, $Recall_i$ (the recall for i -th class) can be defined as:

$$Recall_i = \frac{\tilde{N}_i}{N_i}, \quad (2.58)$$

where \tilde{N}_i is the number of correctly predicted instances for i -th class, N_i is the total number of instances labelled as i -th class. Recall is usually used to evaluate the system's performance for a specific class. When evaluating a general performance of the system for all classes, *weighted average recall* (WAR) (or *accuracy*) is used:

$$\begin{aligned} \text{WAR} &= \sum_{i=1}^{N_c} \lambda_i \text{Recall}_i, \\ \lambda_i &= \frac{N_i}{N}, \end{aligned} \quad (2.59)$$

where λ_i is called the weight for i -th class, N_c is the number of classes, N is the total number of instances.

WAR is widely used because it can give a general evaluation of the performance achieved by the trained system. However, an essential factor ignored by WAR is that, in the real world, data are usually unbalanced in distribution among classes. Thus, the real performance of a recognition system might be overestimated if the correctly classified instances belong to a class that coincidentally occupies a large proportion among the whole instances, and vice versa. Therefore, *unweighted average recall* (UAR) is used as the primary metric in this thesis unless stated otherwise. UAR is defined as:

$$\text{UAR} = \frac{\sum_{i=1}^{N_c} \text{Recall}_i}{N_c}, \quad (2.60)$$

where N_c is the number of classes. It can be easily seen that, WAR will be equal to UAR if the weight λ_i becomes a constant for all classes, i. e., a case of balanced data.

2.7.2 Significance Tests

It is essential to know whether that system A performs better than system B is significant or not in statistics. When comparing the difference of measured values

(e. g., UARs) of two systems, a z -test is adopted. The standard score z can be given as [212]:

$$z = \frac{m_A - m_B}{\sqrt{2m(1 - m)/N}}, \quad (2.61)$$

where $m = (m_A + m_B)/2$, and m_A and m_B are the measure value of system A and system B , respectively, N is the total number of instances. For the one-tailed case (e. g., $m_A > m_B$), the p -value is calculated as:

$$p = 1 - \Phi(z) < \alpha, \quad (2.62)$$

where $\Phi(\cdot)$ denotes the standard normal cumulative distribution function, the α is called the significance level (e. g., .05, .01, .001). Generally, the p -value represents the probability of rejecting the null hypothesis, which means a smaller p -value means a more significant difference between the compared two systems.

When comparing the measures of two systems when they are applied in multiple experiments, e. g., comparing the performances of different features fed into a variety of classifiers, Student's t -test is used. The test static is calculated as:

$$t = \frac{\bar{m}_A - \bar{m}_B}{\sqrt{\frac{\sigma_A^2}{N_A} + \frac{\sigma_B^2}{N_B}}}, \quad (2.63)$$

where \bar{m}_A , \bar{m}_B are the sample means of the measure values from a sample size of N_A and N_B , respectively. σ_A and σ_B are the sample standard deviations. The p -value of the Student's t -test can be calculated from a Student's t -distribution [212].

In this thesis, a one-tailed z -test and a one-tailed Student's t -test are used in different cases (will be stated when appears) to take the significance tests. For theoretical details, one can refer to [212, 213].

AGASC covers a wide range of topics on computational audio analysis using signal processing and machine learning to fulfil the relevant task. In this thesis, three typical general audio signals related respectively to healthcare, ecological monitoring, and public/home security surveillance were proposed in Chapter 1.

In this chapter, comprehensive experiments will be executed to evaluate the approaches proposed in Chapter 2. For snore sound classification, wavelet features are introduced and compared with other widely used acoustical features. Particularly, a comprehensive comparison on features and classifiers is given. Moreover, a BoAW approach is employed to improve the performance achieved by features using functionals. For bird sound classification, data enrichment using AL is investigated on reducing the human expert annotation work. Moreover, the effectiveness and robustness of AL algorithms (i. e., SI-AL and LCS-AL) and selected classifiers (i. e., SVM and KELM) is evaluated. For acoustic scene classification, WPTE and WEF feature sets are combined with the COMPARE feature set by a late fusion strategy. SVM, GRNN, and BGRNN are selected as the classification models. The effectiveness and robustness of proposed systems are measured both in clean and noisy environments.

To facilitate the reproducibility and sustainability of the relevant research, all the databases used in this thesis are publicly accessible. Table 3.1 gives an overview of all the databases used in this thesis.

For each topic involved in this thesis, there will be a background introduced before the experiments. Further, a brief description of each database used is given. In addition, the detailed experimental setup and the corresponding results will be illustrated. A summary will be added at the end of each section.

¹<http://www.animalsoundarchive.org/RefSys/Statistics.php>

Table 3.1: Overview of the three databases used in this thesis.

	Instances (#)	Classes (#)	Time (hours)
MPSSC [80]	828	4	0.35
MNB Bird Sound ¹	5 060	86	4.00
DCASE 2017 Acoustic Scene [48]	6 300	15	17.50

3.1 Computer Audition for Snore Sound Excitation Localisation

Snore sound classification based on its excitation location is a young field with limited databases and literature coverage. As main part of applications in this thesis, the work will be firstly done on comparing features and classifiers. In particular, wavelet features (cf. Section 2.1.6) are introduced and investigated in the area of SnS classification. Furthermore, wavelet LLDs applied within the BoAW approach (cf. Section 2.2.2) can contribute to a better performance than traditional functionals (cf. Section 2.2.1) when using a simple NB classifier.

3.1.1 Background

Obstructive sleep apnea (OSA) is a chronic disease affecting 13% (men) and 6% (women) in the US population [10], which can severely affect health and quality of life. OSA is defined as a sleep disorder with subject’s cessation, or reduction of airflow during sleep due to a complete (*apnoea*) or partial (*hypopnea*) collapse of the upper airway for more than ten seconds (with five or more episodes per hour) [5]. Usually, it is associated with a decrease in oxyhemoglobin saturation [5]. When untreated, OSA increases the risks of stroke [6], hypertension [7], myocardial infarction [8], and is associated with diabetes [214, 215]. Moreover, OSA can be linkable to accidents and harmful for patients’ mood [216, 217]. It is reported that, loud snoring is a typical symptom among more than 80% of OSA patients [218]. Pioneers’ work was focused on analysing acoustic properties of snoring, which aims to develop methods to replace, or complement *Polysomnography* (PSG), the gold standard for diagnosis of OSA [4]. The relevant results are promising and encourage that, methods based on SnS acoustical analysis can reach accuracies of up to 80% and sensitivities and specificities of up to 90% on detection of OSA (small populations between 5 and 60 subjects) [219].

Depending on individual anatomy and multifactorial mechanisms of SnS generation, the surgical options for OSA patients may differ and include, among others, soft palate stiffening, tongue base suspension, hypoglossal nerve stimulation, mandibular advancement, tonsillectomy or tonsillotomy, uvulotomy, uvulopalatopharyngoplasty,

hyoid suspension, and epiglottectomy [11]. Specifically, for a severe OSA patient, multilevel surgery, i. e., a combination of several surgical treatments at different anatomic levels will be used [12]. Therefore, knowing the individual anatomical site of snoring generation and the obstruction mechanism is more important for *ear, nose, and throat* (ENT) surgeons than only screening OSA. Among a variety of methods, *drug induced sleep endoscopy* (DISE) has been increasingly used to identify the location and form of vibrations and obstructions in the upper airway [13]. Nevertheless, the drawback of DISE is obvious. It is time-consuming, costly, straining for subjects, and cannot be performed in a case of natural sleep. Multi-channel pressure measurement might be another option, in which a thin tube with multiple pressure sensors will be introduced into the upper airway of the subject [220–222]. The obstruction location during an apnoeic or hypopnoeic event can be determined via observing the pressure changes during breathing of the different sensors. Even though this method can be used in natural sleep, introducing the tube within the upper airway cannot be tolerated by every subject.

Computer audition analysis of SnS using *signal processing* and *machine learning* can facilitate the study on developing a less-invasive method to plan a targeted ENT surgery not only for OSA [5]) patients, but also for primary snorers (snoring with the absence of apnoeic or hypopnoeic episodes [80]). The early work on this topic was focused on finding the statistical differences between properties of simple features extracted from a few kinds of SnS. The *fundamental frequency* was used to distinguish SnS generated from soft palate, tonsils/tongue base, combined location (both palate, and tonsils/tongue base), and the larynx [223]. It was found by Miyazaki et al. that, the average value of the fundamental frequency was 102.8 Hz, 331.7 Hz, 115.7 Hz, and around 250.0 Hz in the corresponding sites mentioned above, respectively (an examination based on 75 adult subjects) [223]. Hill et al. indicated that the *crest factor*, i. e., the ratios of the peak to the root mean square value of a time-varying signal, was significantly higher for palatal snorers in 11 subjects ($p < .01$, Student's *t* or Mann-Whitney tests) [224]. In addition, *peak frequency*, *centre frequency*, and *power ratio* were investigated by Agrawal et al. in [103] among palate and tongue-based SnS during natural and induced sleep (16 subjects involved). Beeton et al. studied the *statistical dimensionless moment coefficients* of *skewness* and *kurtosis* of palatal and non-palatal SnS were collected from 15 subjects [225]. A series of psychoacoustic features (e. g., *loudness*, *sharpness*, *roughness* and *fluctuation strength* [226]) were studied by Herzog et al. on SnS defined as velar, velar obstructive, tonsillar, and post-apnoeic snoring (based on 41 subjects) [227]. In summary, the studies mentioned above were only concentrated on evaluating some certain acoustic features selected by human experts for their sensitivity to the anatomical mechanisms of snoring sound generation, or the obstruction in the upper airway, whereas more advanced techniques in signal processing and machine learning were not involved.

The pilot work done by Qian et al. introduced *wavelet features* to SnS classifica-

tion by an SVM classifier [18]. Wavelet features were demonstrated to be superior to some other widely-used features like crest factor, fundamental frequency, power ratio, formants, and MFCCs, for the recognition of SnS [18]. Furthermore, Qian et al. investigated a detailed study on comparing multiple acoustic features and classifiers for SnS classification in [21]. In that study, contributions of features like crest factor, power ratio, and fundamental frequency, are very limited [21]. However, these studies were all done within a small group of subjects (less than 50 independent subjects). Besides, the instances used for machine learning were segmented episodes rather than whole snore events [18, 21]. This might cause overestimated results due to a replicated training from too many similar instances for one certain class. In medical practice, it is more often the task to analyse a whole snore event rather than the segmented episodes from it.

In this thesis, a comprehensive comparison of *features* and *classifiers* will be investigated in a publicly accessible SnS database (cf. Section 3.1.2), in which 219 independent subjects were involved. Furthermore, a novel method combining wavelet features and the *bag-of-audio-words* (BoAW) approach is presented. In addition, the best results of the experiments will be compared with the other colleagues' state-of-the-art work by using the same database.

3.1.2 Munich Passau Snore Sound Corpus

The *Munich Passau Snore Sound Corpus* (MPSSC) [80] was first released as a sub-challenge in the INTERSPEECH 2017 COMPUTATIONAL PARALINGUISTICS CHALLENGE (COMPARE) [111]. The MPSSC contains audio clips (16 bit PCM encoded single channel, 16 kHz sampling rate) from selected audio-video recordings taken during DISE from three medical centres, i. e., Klinikum rechts der Isar, Technische Universität München, Germany, Alfried Krupp Hospital, Essen, Germany, and University Hospital Halle (Saale), Germany. Detailed information about the SnS data acquisition system, data selection and labelling can be found in the work by Christoph *et al.* [80]. In the audio-track of the DISE videos, snore events were separated using a combination of automated and manual selection steps [80]. The selected snore events were then labelled by an ENT expert by watching the DISE videos and based on the VOTE classification [228] ('V' represents *the level of the velum*; 'O' represents *the oropharyngeal area*; 'T' represents *the tongue base*; 'E' represents *the level of the epiglottis*). Only the snore events which showed one clear vibration source were included in the corpus, the ones with mixed or unclear source of vibration were excluded. Figure 3.1 shows the typical screen shots of the corresponding vibration locations from the DISE videos.

The final MPSSC database contains 828 snore events from 219 independent subjects (93.6% are male). The overall time duration of MPSSC is 1 250.11 s (approximately 0.35 hours), and the average length of the events is 1.51 s (ranging from 0.73 s to 2.75 s). Balanced by class, centre, gender, and age, the whole database

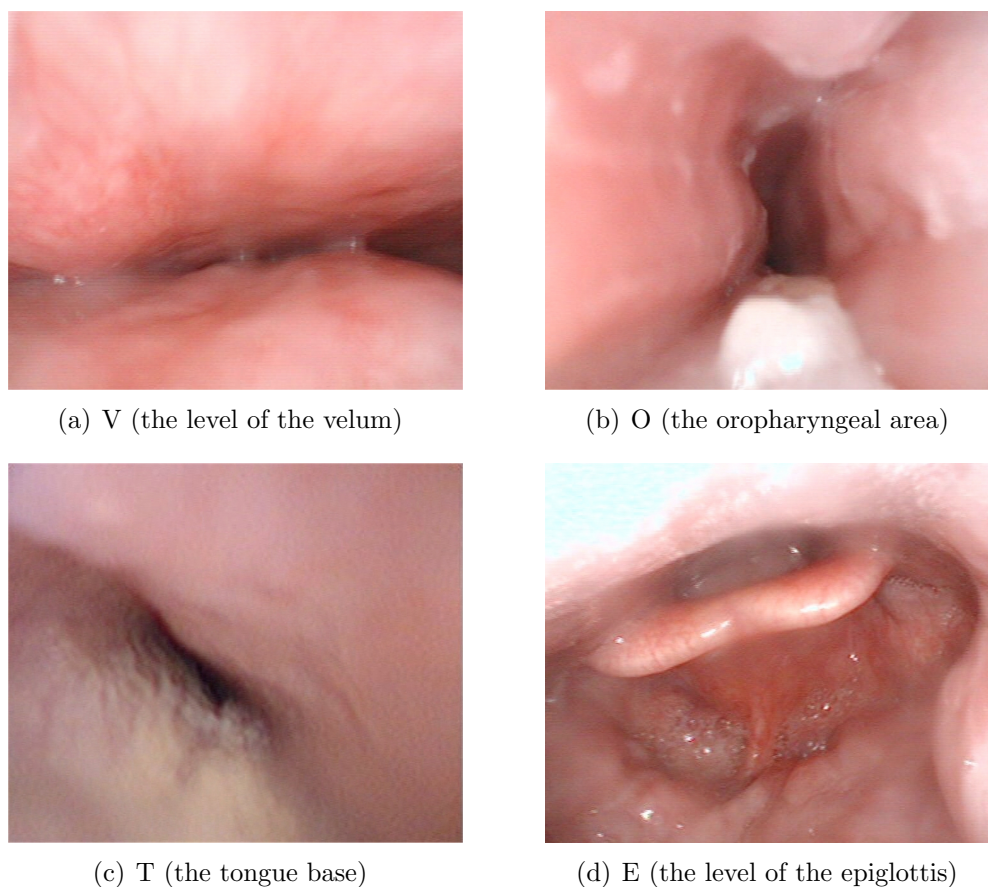


Figure 3.1: Typical screen shots taken from the DISE video recordings showing snoring from soft palate (velum) (V), oropharyngeal (O), tongue base (T), epiglottal (E).

was partitioned into a *train*, a *development*, and a *test* set. For details of the data partition, readers can refer to [80]. Table 3.2 illustrates the number of snore events per class in each partitioned split. For the sake of comparability, this thesis uses the same splits as in the INTERSPEECH 2017 COMPARE Snoring sub-challenge [111]. It can be seen that, MPSSC is an extremely unbalanced database, in which the instances with class of V occupy the maximum proportion (58.5%), then comes the class of O (26.1%). The T type and E type of instances only account for 4.7% and 10.8%, respectively. However, as indicated in [80], such an unbalanced character of SnS is in line with some early findings by DISE examinations [229].

Figure 3.2 illustrates the examples of *waveforms* and *spectrograms* of the different types of snore events. From the time domain analysis, SnS belongs to typical non-stationary signals. From the frequency domain analysis, the main energy components in SnS are concentrated in the low frequency area (below around 4 kHz in

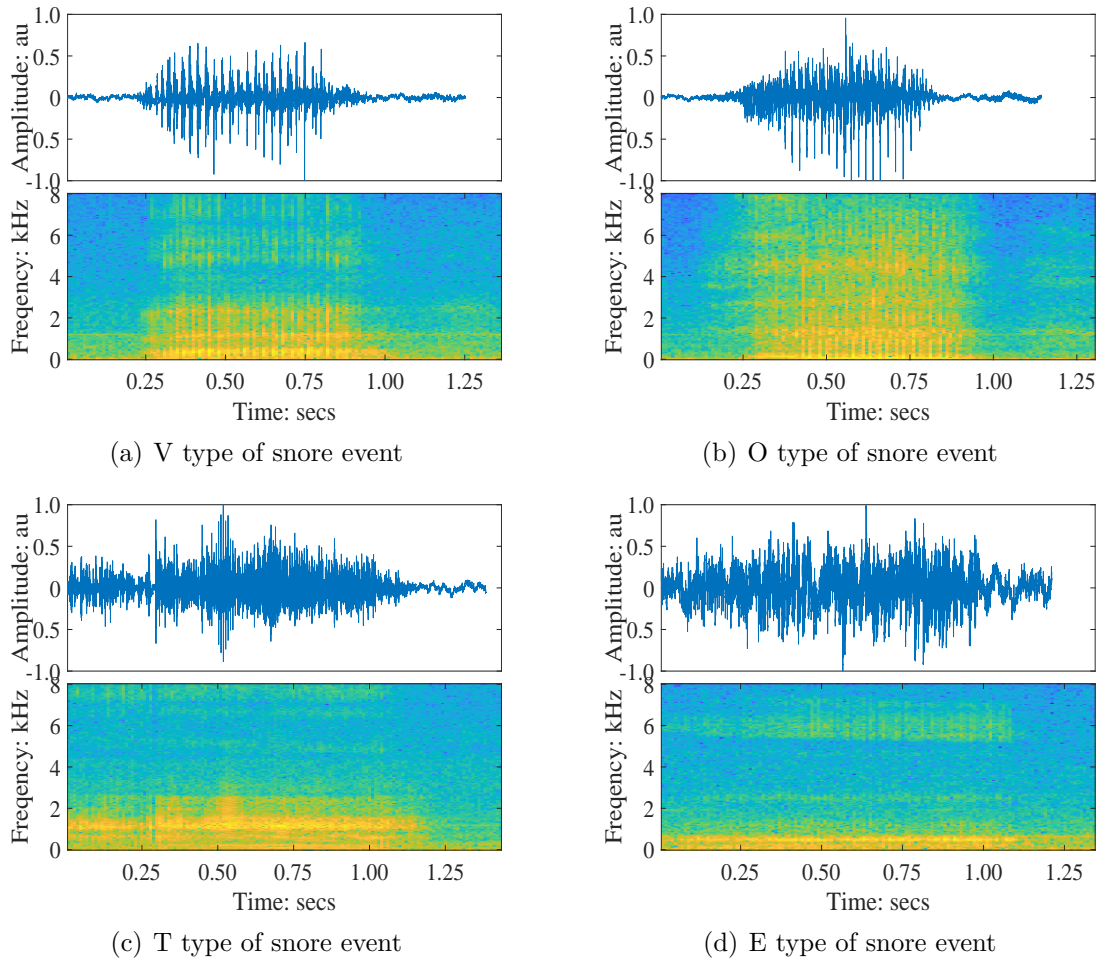


Figure 3.2: Examples of waveforms (top row) and spectrograms (bottom row) for the snore event labelled as type of V, O, T, and E. The waveforms are normalised and the amplitude has an arbitrary unit (au).

the four examples given in Figure 3.2).

3.1.3 Experimental Setup

Acoustic Features

Based on previous studies [21, 93], Formants, SFFs, SERs, MFCCs, WTE, WPTE, and WEF are investigated and compared in this thesis. The definition of LLDs of those features mentioned above can be found in Section 2.1. The first three formant frequency values (F1, F2, and F3) are included in the feature set of Formants. The MFCCs (0–12) are obtained from 27 triangular Mel filter banks added to the analysed chunk (frame) of SnS. The subband for extracting SFFs and SERs is selected

Table 3.2: The number of snore events per class in each partitioned split, as originally used in [111]. Dev: development. Details of data partition on *class*, *centre*, *gender*, and *age* can be found in [80].

	Train	Dev	Test	Σ
V	168	161	155	484
O	76	75	65	216
T	8	15	16	39
E	30	32	27	89
Σ	282	283	263	828

Table 3.3: The frame size and overlap for extracting LLDs following with the dimension of LLDs. Dim: dimension.

	Frame Size	Overlap	Dim of LLDs
Formants	16 ms	12 ms	3
SFFs	32 ms	16 ms	19
SERs	32 ms	8 ms	16
MFCCs	32 ms	24 ms	13
COMPARE ²	20 ms	10 ms	65
WTE	16 ms	4 ms	16
WPTE	32 ms	16 ms	1 023
WEF	64 ms	32 ms	87

as 500 Hz empirically in initial experiments. In addition, the COMPARE feature set (cf. Section 2.1.5) is involved in the study. It was observed that the *frame size* and *overlap* of the analysed audio chunk for extracting LLDs can effect the final classification performance [20, 93]. In this thesis, based on tremendous experiments in previous study [93], the frame size and overlap was empirically set as Table 3.3 shows. The COMPARE feature set is extracted by OPENSIMILE toolkit [109, 110]. The wavelet types and the corresponding maximum decomposition level J_{max} for WTE, WPTE, and WEF are chosen based on initial experiments, which are listed in Table 3.4. To get rid of the effects by data imbalance and outliers, the data (features) are upsampled and standardised before feeding into the classification models.

²The frame size and overlap to extract LLDs of F0 is set to 60 ms and 50 ms, respectively.

³The short names representing the wavelet types can be referred to the MATLAB Wavelet Toolbox: <https://www.mathworks.com/products/wavelet.html>.

Table 3.4: The parameters for extracting wavelet features in the task of snore sound classification. J_{max} : maximum decomposition level.

	Wavelet Type ³	J_{max}
WTE	‘bior2.8’	3
WPTE	‘haar’	9
WEF	‘coif5’	5

Classification Models

For classification, the classical models (cf. Section 2.4.1), e. g., NB, k -NN, SVM, and RF are involved. Besides, deep learning models (cf. Section 2.4.2), MLP and SAE, and extreme learning models (cf. Section 2.4.3), ELM and KELM, are investigated. However, recurrent neural networks (cf. Section 2.4.2), GRNN and BGRNN are not used due to the limited context information in such short snore events (less than 3 seconds). The NB and SVM classifiers are implemented by the WEKA toolkit [209] and the other classifiers mentioned above are implemented by MATLAB. When training the MLP and SAE classifiers, the fast and efficient *scaled conjugate gradient* (SCG) method [176] is used. The main parameters of each classifier are tuned and optimised within a searching grid on the development set, and applied to the model for evaluating the test set (see Table 3.5).

BoAW Approach

For the processing of the BoAW approach, the toolkit OPENXBOW is used. The frame size and overlap for extracting LLDs from chunks of each feature set are shown in Table 3.3. As the counterpart of BoAW-based features, the functionals applied to LLDs of COMPARE can be found in Table 2.2. The *functionals* for the other kinds of LLDs are selected as *maximum value*, *minimum value*, *arithmetic mean*, and *linear regression offset*, which were demonstrated to be efficient in previous studies [21]. The *codebook size* and *number of assignments* are empirically optimised to be 5 000 and 10, respectively. The codebook is generated via a random sampling process (by default random seed of OPENXBOW). Both the LLDs and the resulting term-frequency histograms are standardised. In experiments on BoAW, the NB classifier is chosen because it was found to be superior for BoAW features to other models in initial experiments such as SVM or MLP, which might explain by, Naïve Bayes could be much less prone to overfitting than others when learning BoAW features from such a small size of SnS data.

⁴<https://www.mathworks.com/help/stats/classificationknn-class.html>

⁵<https://www.mathworks.com/help/stats/treebagger.html>

Table 3.5: The searching grids for tuning the main parameters of each classifier. These parameters will be optimised on the development set, and applied to the test set.

Classifier	Main Parameters
NB	estimator: kernel density, normal distribution
k -NN	k -value: $\{1, 10, 20, \dots, 90, 100\}$; distance metrics ⁴ : ‘euclidean’, ‘cityblock’, ‘chebychev’, ‘cosine’, ‘correlation’, ‘hamming’, ‘jaccard’, ‘minkowski’, ‘spearman’
SVM	linear kernel; C_s -value: $\{10^{-6}, 10^{-5}, \dots, 10^5, 10^6\}$
RF	number of trees: $\{2^1, 2^2, \dots, 2^9, 2^{10}\}$; fraction for the treebagger ⁵ : $\{0.1, 0.2, \dots, 0.9, 1.0\}$
ELM	activation functions: {‘sigmoid’, ‘sine’, ‘hardlim’, ‘tribas’, ‘radbas’}; number of hidden units: $\{2^1, 2^2, \dots, 2^{14}\}$ C_e -value: $\{10^{-6}, 10^{-5}, \dots, 10^5, 10^6\}$
KELM	linear kernel; C_e -value: $\{10^{-6}, 10^{-5}, \dots, 10^5, 10^6\}$
MLP	two hidden layers; number of hidden units: $\{2^1, 2^2, \dots, 2^9, 2^{10}\}$
SAE	an architecture of two-layer stacked auto-encoders; number of hidden units: $\{2^1, 2^2, \dots, 2^9, 2^{10}\}$; L_2 weight regularisation: $\{10^{-5}, 10^{-4}, \dots, 10^4, 10^5\}$; sparsity proportion: $\{0.1, 0.2, \dots, 1.0\}$

3.1.4 Features and Classifiers for Snore Sound Classification

The SnS classification results by multiple features and classifiers are listed in Table 3.6. Additionally, the significance levels by one-tailed Student’s t -test on features and classifiers are given in Table 3.7 and Table 3.8, respectively. The features are fed into each classifier in the form of functionals (cf. Section 2.2.1).

As to features, wavelet features i. e., WTE, WPTE, and WEF are demonstrated to be efficient in this study (see Table 3.7). This is consistent with the previous findings in [18]. The experiments showed the excellent ability of wavelet transformation in analysis of SnS, a kind of non-stationary signals. Moreover, it is reasonable to think that SnS might be distinguishable by the energy distribution differences between subbands. Therefore, the multi-resolution analysis by wavelet features can contribute to find more suitable subband representations than simple STFT-based ones (e. g., SERs). As the standard and popular feature sets, MFCCs and COMPARE have a comparable performance with wavelet features. Previously, formants were widely investigated for their clear relationship between the properties and the anatomical structure of the upper airway [95–97]. However, formants cannot perform so well in this study. Compared with other feature sets, Formants have a very

3. Applications

Table 3.6: Classification results (UARs: [%]) achieved by multiple features and classifiers. Parameters are optimised on the development set and applied to the test set. Results of the development set are the ones achieved by the optimised model. Dev: development. The results higher than the official baseline (58.5% in UAR, refer to [111]) of the INTERSPEECH 2017 COMPARE Snoring sub-challenge are highlighted (bold).

		NB	k -NN	SVM	RF	ELM	KELM	MLP	SAE
Formants	Dev	31.9	42.9	35.6	38.2	43.8	32.2	40.0	38.0
	Test	49.3	35.6	39.3	33.0	30.4	39.9	34.0	43.5
SFFs	Dev	32.5	43.6	40.6	41.2	42.5	38.7	43.7	45.1
	Test	49.2	47.6	45.1	44.9	29.4	50.5	40.6	46.8
SERs	Dev	29.7	41.7	38.5	38.7	46.0	37.7	45.4	39.0
	Test	43.4	47.9	56.6	34.9	34.6	56.4	25.3	42.1
MFCCs	Dev	30.0	54.4	35.7	46.9	45.8	38.0	50.5	50.6
	Test	45.7	45.2	54.5	52.7	48.8	48.8	54.4	43.8
COMPARE	Dev	28.6	44.0	36.0	42.8	43.5	39.3	51.6	43.9
	Test	27.4	48.7	54.6	55.0	49.3	51.1	53.5	39.1
WTE	Dev	27.7	45.2	46.2	38.9	49.9	46.4	47.6	46.8
	Test	49.9	38.5	60.3	58.1	52.7	46.0	48.8	46.0
WPTE	Dev	34.0	46.7	42.3	42.9	42.7	44.5	49.8	43.7
	Test	46.6	45.5	62.2	52.3	61.5	59.2	47.8	50.7
WEF	Dev	33.5	49.4	40.7	42.3	36.2	48.0	41.4	45.5
	Test	49.7	57.7	53.8	37.2	63.5	51.5	58.1	44.8

limited dimension in LLDs (see Table 3.3), which result in a limited performance in expressing more information in the spectrum to classify SnS. Similarly, SFFs yields to other excellent feature sets (e.g., wavelet features) in this study. The reason might be the lower amount of information carried by the frequency-based features than by the more sophisticated wavelet features. Specifically, COMPARE has a limited performance in this study. As a large feature set designed originally for the computational paralinguistics challenge tasks [108], COMPARE has many redundant LLDs, which makes it not perfectly suitable for SnS classification. On classifiers, SVM shows its significant superiority over NB, k -NN, and SAE (see Table 3.8). As a popular and standard machine learning technique, SVM were regarded as the standard classifier for many benchmarks in past decades. In contrast, the deep learning models are restrained by the limited size of SnS data in this study.

Table 3.7: Significance levels obtained from the statistical comparison (one-tailed Student’s t -test) by measuring UARs on the test set between different features by feeding them into multiple classifiers. The comparisons are made between a pair of counterparts listed in the left column and the top row of the table. The significance levels are highlighted by grayscale shading, based on the values $p < .05$, $p < .01$, and $p < .001$.

Sign. Levels	Formants	SFFs	SERs	MFCCs	CoMPARE	WTE	WPTE	WEF
Formants								
SFFs								
SERs								
MFCCs								
CoMPARE								
WTE								
WPTE								
WEF								

■	$p < .05$	■	$p < .01$	■	$p < .001$
---	-----------	---	-----------	---	------------

The best four results reaching an UAR at 63.5 %, 62.2 %, 61.5 %, and 60.3 % are achieved by the models of WEF-ELM, WPTE-SVM, WPTE-ELM, and WTE-SVM, respectively. Table 3.9 shows the confusion matrices of the four models. Commonly, ‘E’ (the epiglottis) type of SnS is easier to be classified than the other three types, whereas the ‘O’ (the oropharyngeal area) type of SnS can be the most difficult one to be recognised. More precisely, the ‘O’ type of snores can be wrongly classified as ‘V’ type (the level of the velum), which could be reasonably explained by the acoustical similarities of the two types of SnS due to the closed anatomical position in the upper airway of the two locations (refer to [21]). For ‘T’ (the tongue base) type of SnS, the four models share a common recall at 68.8 %. The best recall for ‘V’ (71.6 %), ‘O’ (32.3 %), and ‘E’ (96.3 %) can be achieved by the models of WEF-ELM, WPTE-ELM, and WTE-SVM, respectively. Interestingly, when being fed with the same feature set (WPTE), SVM and ELM share a similar confusion matrix (see Table 3.9(b) and Table 3.9(c)).

Even though all of the four models beat the official baseline (58.5 % in UAR) of the INTERSPEECH 2017 CoMPARE Snoring sub-challenge [111], the results

Table 3.8: Significance levels obtained from the statistical comparison (one-tailed Student’s t -test) by measuring UARs on the test set between different classifiers fed by multiple features. The comparisons are made between a pair of counterparts listed in the left column and the top row of the table. The significance levels are highlighted by grayscale shading, based on the values $p < .05$, $p < .01$, and $p < .001$.

Sign. Levels	NB	k -NN	SVM	RF	ELM	KELM	MLP	SAE
NB								
k -NN								
SVM								
RF								
ELM								
KELM								
MLP								
SAE								

$p < .05$
 $p < .01$
 $p < .001$

are not significantly higher ($p > .05$, one-tailed z -test). Table 3.10 lists the results of the late fusion, in which the *majority voting* (MV) and *margin sampling voting* (MSV) strategies are adopted (cf. Section 2.6). There are three results (by the MV strategy) showing a significance level of $p < .05$ (one-tailed z -test) when comparing them with the official baseline. In particular, the best two results (67.1% and 66.4%) are comparable to the results achieved by training complicated deep neural networks in the work [163] (67.0% in UAR) and the work [164] (66.5% in UAR). Furthermore, even the third place result, i. e., an UAR of 65.5% beats the winner’s performance (a UAR of 64.2%, refer to [230]) in the sub-challenge. The confusion matrices of the three results are shown in Table 3.11. It can be found that, the recalls of the ‘V’ and ‘E’ types of SnS are improved as compared with the single models (see Table 3.9). Nevertheless, the fusion has limited positive effect to recalls of the ‘O’ and ‘T’ types of SnS.

3.1.5 A Bag of Acoustic Features for Snore Sound Classification

In this section, the features of SnS mentioned previously will be investigated and compared by the method of *Functionals* (cf. Section 2.2.1) and *BoAW* (cf. Sec-

3.1. Computer Audition for Snore Sound Excitation Localisation

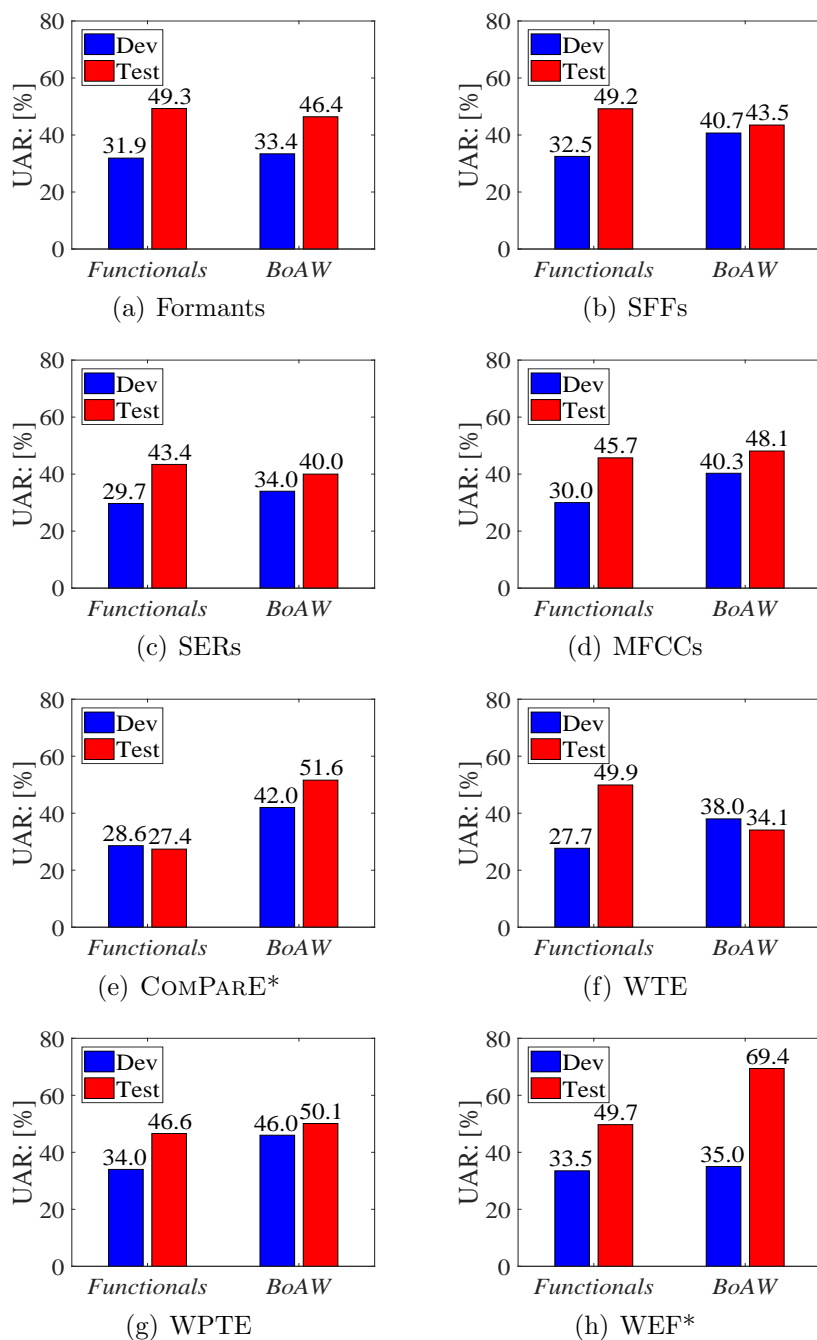


Figure 3.3: Results (UARs: [%]) achieved by each acoustic feature set within *Functionals* and *BoAW*. Feature sets showing significant improvement ($p < .001$, one-tailed z-test) by *BoAW* compared with *Functionals* on the test set are marked by an asterisk.

3. Applications

Table 3.9: Confusion matrices (normalised, in [%]) of the best four models on the test set. The diagonal elements are the recalls corresponding to each class.

(a) WTE-SVM					(b) WPTE-SVM				
<i>Pred -></i>	V	O	T	E	<i>Pred -></i>	V	O	T	E
V	60.6	3.9	7.7	27.7	V	61.9	22.6	7.1	8.4
O	35.4	15.4	3.1	46.2	O	55.4	29.2	10.8	4.6
T	0.0	0.0	68.8	31.3	T	25.0	6.3	68.8	0.0
E	3.7	0.0	0.0	96.3	E	11.1	0.0	0.0	88.9

(c) WPTE-ELM					(d) WEF-ELM				
<i>Pred -></i>	V	O	T	E	<i>Pred -></i>	V	O	T	E
V	56.1	22.6	11.0	10.3	V	71.6	6.5	13.5	8.4
O	50.8	32.3	10.8	6.2	O	49.2	24.6	15.4	10.8
T	18.8	6.3	68.8	6.3	T	6.3	0.0	68.8	25.0
E	7.4	3.7	0.0	88.9	E	0.0	0.0	11.1	88.9

tion 2.2.2), respectively. The classification model is NB as it was found to be efficient for this study in initial experiments. Figure 3.3 illustrates the classification results of SnS by each kind of feature set within *Functionals* or *BoAW*. Generally, for the development set, all of the feature sets can have an improvement in UAR when using *BoAW* instead of *Functionals*. For the test set, MFCCs, COMPARE, WPTE, and WEF reach a better UAR using *BoAW* while Formants, SFFs, SERs, and WTE lead to a decrease in performance. When paying more attention to the results on the test set, the improvements by *BoAW* are significant on the COMPARE feature set and WEF ($p < .001$, one-tailed z -test).

On *Functionals*, WTE and WEF perform best with a UAR of 49.9%, and 49.7%, respectively. Furthermore, Formants (UAR of 49.3%) and SFFs (UAR of 49.2%) show comparable performance to the two aforementioned wavelet features when using *Functionals*. The LLDs of COMPARE yield to others when applied with *Functionals*, only reaching a UAR of 27.4%.

On *BoAW*, WEF and COMPARE achieve the best results among all feature sets, with a UAR of 69.4%, and 51.6%, respectively. MFCCs are comparable to WPTE (UAR: 48.1% vs 50.1%). In particular, LLDs of COMPARE considerably improve the performance from 27.4% to 51.6% UAR ($p < .001$, one-tailed z -test) when using *BoAW* rather than *Functionals*. This indicates that the COMPARE feature set can be enhanced to be more suitable to describe the inherited characteristics of SnS with the help of *BoAW* when using an NB classifier. In addition, the best feature set,

3.1. Computer Audition for Snore Sound Excitation Localisation

Table 3.10: Results (UARs: [%]) achieved by late fusion. The symbol \times denotes the model involved in the late fusion. Dev: development. MV: majority voting. MSV: margin sampling voting.

	WTE-SVM	WPTE-SVM	WPTE-ELM	WEF-ELM	Dev	Test
MV	\times	\times			44.7	60.6
		\times	\times		41.1	61.9
		\times		\times	38.6	62.9
	\times		\times		48.5	60.5
	\times			\times	45.6	62.4
			\times	\times	38.3	63.3
	\times	\times	\times		42.1	62.0
	\times	\times		\times	39.6	67.1
		\times	\times	\times	41.4	62.0
		\times	\times	\times	42.2	66.4
MSV	\times	\times			40.9	65.5
	\times	\times			45.6	61.6
		\times	\times		43.8	62.2
		\times		\times	42.1	62.4
	\times		\times		47.7	60.7
	\times			\times	45.6	60.1
			\times	\times	40.5	62.8
	\times	\times	\times		46.2	61.3
	\times	\times		\times	45.0	60.7
		\times	\times	\times	43.8	62.4
	\times		\times	47.7	60.7	
	\times	\times	\times	45.5	60.2	

i. e., WEF with *BoAW*, reaches a UAR of 69.4%, which improves by 19.7% from the baseline by *Functionals* ($p < .001$, one-tailed z -test). *BoAW* can provide a global view on the statistical information of LLDs from the whole data set, whereas the statistical information achieved by *Functionals* is limited (to one instance).

The confusion matrices of the results on the test set for COMPARE and WEF with *Functionals* and *BoAW* are presented in Table 3.12. One common finding, both for COMPARE and WEF, is that *BoAW* decreases the recall on recognition of ‘V’ type snores. Nevertheless, for ‘T’ type snores, *BoAW* can dramatically improve the recall for COMPARE (from 0.0% to 43.8%, $p < .001$, one-tailed z -test), and WEF (from 12.5% to 75.0%, $p < .001$, one-tailed z -test), which is the main contribution to the improvement in UAR. In particular, for COMPARE, the recall of recognising

3. Applications

Table 3.11: Confusion matrices (normalised, in [%]) of the best three models on the test set by late fusion. The diagonal elements are the recalls corresponding to each class.

(a)					(b)				
<i>Pred -></i>	V	O	T	E	<i>Pred -></i>	V	O	T	E
V	74.8	7.7	8.4	9.0	V	72.3	5.8	11.0	11.0
O	52.3	24.6	9.2	13.8	O	50.8	24.6	9.2	15.4
T	6.3	0.0	68.8	25.0	T	6.3	0.0	68.8	25.0
E	0.0	0.0	0.0	100.0	E	0.0	0.0	0.0	100.0

(c)				
<i>Pred -></i>	V	O	T	E
V	71.6	8.4	11.0	9.0
O	52.3	29.2	9.2	9.2
T	18.8	6.3	68.8	6.3
E	7.4	0.0	0.0	92.6

‘E’ type snores has been improved from 3.7% to 74.1% ($p < .001$, one-tailed z -test), which results in another considerable enhanced performance for the final UAR. For the recognition of ‘O’ type snores, COMPARE and WEF respectively show an increase of 21.5% ($p < .01$, one-tailed z -test), and 27.7% ($p < .001$, one-tailed z -test) for recall after using *BoAW* instead of *Functionals*. Thus, it can be concluded that, for some rare types of SnS, i. e., ‘O’, ‘T’ and ‘E’, *BoAW* can improve their recalls compared with *Functionals*, which will be beneficial to the unbalanced distribution of SnS data. The types ‘V’ and ‘O’ are still the most misclassified samples (both for *Functionals* and *BoAW*). Most probably, this is due to the small sample size in the database, a limitation that should be targeted in future work.

3.1.6 Summary

Computer audition for localisation of snore sound excitation can facilitate the development of less-invasive methods to plan a targeted ENT surgery for both OSA patients and primary snorers. However, the research in this field is very limited. Section 3.1.1 briefly introduced the background of relevant work. A publicly accessible SnS database was described in Section 3.1.2. A comprehensive comparison of features and classifiers was presented in Section 3.1.4. Four best models trained by SVM or ELM within wavelet features were demonstrated to be superior to other models in SnS classification. Moreover, a late fusion of the four models can generate

Table 3.12: Confusion matrices (normalised, in [%]) of COMPARE and WEF with *Functionals* and *BoAW* on the test set. The diagonal elements are the recalls corresponding to each class.

(a) COMPARE: <i>Functionals</i>					(b) COMPARE: <i>BoAW</i>				
<i>Pred -></i>	V	O	T	E	<i>Pred -></i>	V	O	T	E
V	70.3	22.6	2.6	4.5	V	31.6	47.1	6.5	14.8
O	55.4	35.4	7.7	1.5	O	21.5	56.9	4.6	16.9
T	56.3	37.5	0.0	6.3	T	50.0	0.0	43.8	6.3
E	22.2	66.7	7.4	3.7	E	3.7	18.5	3.7	74.1

(c) WEF: <i>Functionals</i>					(d) WEF: <i>BoAW</i>				
<i>Pred -></i>	V	O	T	E	<i>Pred -></i>	V	O	T	E
V	61.3	32.3	1.3	5.2	V	49.7	39.4	5.2	5.8
O	40.0	40.0	1.5	18.5	O	18.5	67.7	1.5	12.3
T	43.8	43.8	12.5	0.0	T	25.0	0.0	75.0	0.0
E	7.4	7.4	0.0	85.2	E	3.7	11.1	0.0	85.2

a best result reaching 67.1% UAR, which significantly outperformed the official baseline (58.5% of UAR) of the INTERSPEECH 2017 COMPARE Snoring sub-challenge ($p < .05$, one-tailed z -test). Section 3.1.5 proposed a bag of acoustic features which used BoAW (cf. Section 2.2.2) approach instead of functionals (cf. Section 2.2.1) to generate the representations from LLDs for machine learning. The results showed a significant improvement for WEF and the COMPARE feature sets when using BoAW instead of functionals when fed into a NB classifier ($p < .001$, one-tailed z -test). More precisely, a bag of wavelet features (WEF+BoAW) can reach a UAR of 69.4% on the test set, which contributed to the best result for SnS classification in this thesis.

Table 3.13 shows the results of methods which beat the official baseline of the INTERSPEECH COMPARE Challenge 2017 Snoring sub-challenge [111]. The best two methods proposed in this thesis outperform all these models. Gosztolya et al. extracted frame-level features, e.g., MFCCs, voicing probability, harmonics to noise ratio (HNR), fundamental frequency (F0), zero-crossing rate (ZCR) and the combination with derivatives, mean and standard deviation to train an SVM classifier [231]. The final prediction is based on a late fusion of the aforementioned SVM and another SVM model trained on the COMPARE feature set, which reached a UAR of 64.0% on the test set [231]. The winning submission by Kaya et al. took the unbalanced nature of SnS database into account, in which, weighted kernel partial least

3. Applications

Table 3.13: Results (UARs: [%]) of the methods which beat the official baseline of the INTERSPEECH COMPARE challenge 2017 Snoring sub-challenge on the test set. † marks the two submissions without participation in the challenge; ‡ marks the submission of the winner in the sub-challenge. The two best methods proposed in this thesis are highlighted (bold). All the listed SVM classifiers used a *linear* kernel.

	UAR [%]	Main Methods
<i>Official Baseline</i> [111]	58.5	COMPARE Features, SVM
Amiriparian et al. [163]†	67.0	CNN-based Spectrum Features, SVM
Freitag et al. [164]†	66.5	CNN-based Spectrum Features, Evolutionary Feature Selection, SVM
Gosztolya et al. [231]	64.0	COMPARE Features, Voicing Probability MFCCs, HNR, F0, ZCR, SVM
Kaya et al. [230]‡	64.2	COMPARE Features, MFCCs, RASTA-PLP, Fisher Vector, WKPLS, WKELM
Method A	67.1	WTE, WPTE, WEF SVM, ELM, Late Fusion (MV)
Method B	69.4	WEF, BoAW, NB

squares (WKPLS) [230, 232] and a weighted kernel-based extreme learning machine (WKELM) [233, 234] were implemented [230]. Further, LLDs like MFCCs, RASTA-PLP (representations relative spectra perceptual linear prediction) were fused and represented by a Fisher vector [235]. Their best result reached 64.2% UAR by fusing multiple models [230]. Amiriparian et al. proposed deep representations learnt from the spectrum of SnS by convolutional neural networks (CNNs) [160, 162, 236], which achieved a UAR of 67.0% [163]. Freitag et al. [164] used an evolutionary feature selection algorithm based on competitive swarm optimisation [237, 238] to reduce the dimension of features extracted by CNN-based spectrum representations proposed in [163]. Finally, they used approximately 54.8% of the original CNN-based descriptors in [163] to reach a UAR of 66.5% [164].

Generally, sophisticated features are essential for the final performance of the model. For instance, wavelet features (used in this thesis), and deep representations [163, 164] can contribute to an excellent recognition performance using a simple classifier, e. g., SVM or ELM. However, due to the extremely limited number of SnS instances, directly using deep neural networks cannot come up with a good

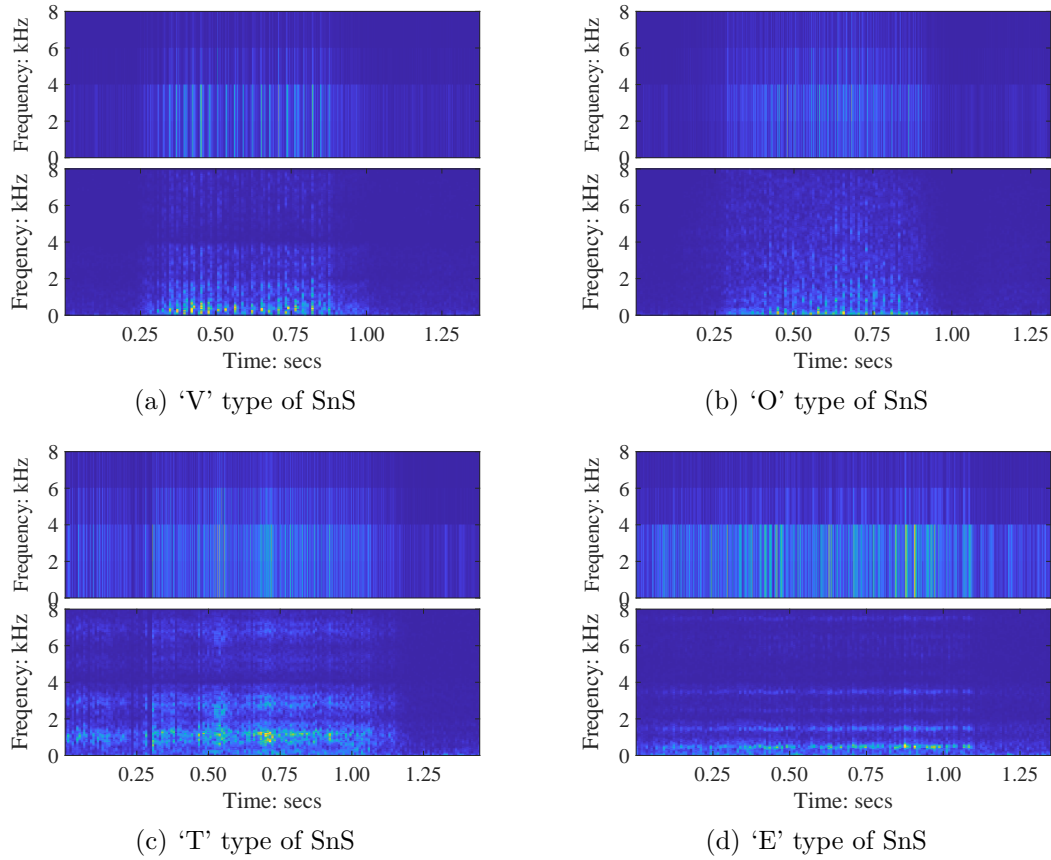


Figure 3.4: Examples of multi-resolution time-frequency analysis of four types of SnS by WPT (top row: $J = 2$, bottom row: $J = 7$). Wavelet type: ‘haar’. J : decomposition level. The audio examples are the same as used in Figure 3.2.

result in these experiments. In particular, wavelet features are found to be efficient both applied with functionals and BoAW. It is reasonable to think that, when using different decomposition levels, a multi-resolution time-frequency analysis of the signal will be given by WT or WPT. Therefore, an abundance of information inherited of the SnS from different resolutions can be included in the wavelet features, which results in an efficient performance for pattern recognition. Figure 3.4 shows a multi-resolution time-frequency analysis by WPT (more suitable than WT for time-frequency analysis) for different types of SnS. One direction of future work can be finding suitable frequency bands for analysis of SnS, which might contribute to more sophisticated wavelet features designed for SnS. In addition, combining wavelet features with some other efficient features (e. g., the CNN-based descriptors [163, 164]) using feature selection methods could better the performance of SnS classification.

3.2 Data Enrichment for Bird Sound Classification

In real-world applications for classification of animal sounds, e.g., bird sounds, there is an unavoidable challenge that human expert annotated audio recordings are much less than the ones unlabelled. Besides, asking human experts to annotate large amounts of bird sound data will be extremely time-consuming, expensive and even difficult to be fulfilled. To address this issue, two *active learning* (AL) algorithms, i.e., SI-AL and LCS-AL, will be investigated and compared in this study. Furthermore, a comparison between two popular classifiers, i.e., SVM and KELM, will also be presented to demonstrate that, changing a classifier in AL's paradigm could considerably enhance the algorithm's performance. The background of bird sound classification will be firstly given in Section 3.2.1. Then, Section 3.2.2 introduces the publicly accessible bird sound database used in this thesis. The experimental setup will be described in Section 3.2.3. The experiments in Section 3.2.4 demonstrate the effectiveness of AL for reducing the human annotation work. To make a comparison of the proposed algorithms in terms of robustness, more experiments will be given in Section 3.2.5. Finally, Section 3.2.6 briefly summarises this study.

3.2.1 Background

Bird sound can offer a plethora of information for helping human experts understand the bird mating and evolutionary changes [239]. Recognising bird species by their sounds will facilitate the development of acoustic-based long-term, non-human systems for monitoring bird species, which can benefit measuring the state of nature [22], tracking climate change [23], and assessing biodiversity within local ecosystems [24, 25].

Looking over the past two decades, there have been increasing efforts of ornithologists, ecologists, and engineers in both *signal processing* and *machine learning* to work collaboratively towards applications for automatically classifying bird sound based only on audio recordings. McIlraith et al. [26] proposed two-layer perceptrons to classify 6 species of birds, which reached correct identification ranging from 82% to 93%. A parametric model for bird sound classification was given by Somervuo et al. [27], by which the average recognition accuracy for single syllables was between 40–50% for 14 common North-European *Passerine* bird species. Chen et al. studied a spectral peak track method that achieved 95% recognition accuracy in noisy environments for 12 natural bird species, and 16 synthesized syllables [28]. Fagerlund used Mel-cepstrum parameters and low-level signal parameters within a decision tree based on *support vector machine* (SVM) classifiers to reach an accuracy of up to 91%, and 98% for 6, and 8 differing species, respectively [29]. A method on using wavelet packet decomposition to extract features was studied by Selin et al. [30], in

which an unsupervised *self-organising map* (SOM) and a supervised *multilayer perceptron* (MLP) were chosen as classifiers. Their reported accuracy reaches to 96 % accuracy for 8 bird species [30]. Lee et al. [31] proposed a method based on two-dimensional cepstral coefficients combined with *Gaussian mixture models* (GMMs) and *vector quantisation* (VQ) to correctly classify nearly 84 % of syllable-based units from 28 bird species. A further work based on [31] uses a novel feature set extracted from spectrogram image shapes was presented in [32], which achieved approximately up to 95 % recognition accuracy among 28 bird species. The methods of frequency track extraction and tonal-based features were studied in [33], and [34], respectively. A *multi-instance multi-label* (MIML) framework for classification of multiple simultaneous bird species was proposed by Briggs et al. [35]. Jančovič et al. [36] proposed an automated bird sound recognition system based on frequency track features and *hidden Markov models* (HMMs) within a decision making by penalised maximum likelihood, which was demonstrated to be efficient for both the case of single bird species, and the case of multiple bird species.

Another direction was specifically given for the detection of syllables from bird sound recordings. A comparative study on *dynamic time warping* (DTW) and HMMs for bird song element recognition within recordings was done by Kogan et al. [42]. It was shown that HMMs needed more training examples than DTW templates whereas DTW-based techniques require expert knowledge to select suitable templates [42]. Ranjard et al. proposed a method based on evolving neural networks for unsupervised bird sound syllable classification [43], in which, a DTW-based distance measure was designed to give an insight into the relationship of spectrogram structures between syllables. An SVM model (with a linear kernel) was reported to achieve around 99 % recognition accuracy in day-long recordings ($26\,055.8 \pm 17\,672.3$ syllables) [44]. Tan et al. introduced an algorithm based on DTW and sparse representation to classify up to 81 phrase classes of *Cassin's Vireo* (*Vireo Cassinii*) [45]. In their study, classification accuracies of 94 % and 89 % on manually and automatically segmented phrases were reached, respectively, using only limited training data (one to five samples per phrase) [45]. A template-based algorithm using DTW and prominent (high-energy) time-frequency regions of training spectrograms was studied in [240], which can outperform DTW and HMMs in most training and test conditions. Moreover, the robustness of this method was demonstrated with data sets of limited sizes, and within noisy background conditions.

Recently, the LifeCLEF Bird task [241] provided the research community a bird sound study on a public database within large scale of bird species, which included 501 species within 14 027 audio recordings in 2014, extended to 1500 species with 36 496 audio recordings in 2017. A method on a combination of large scale feature sets (with feature selection), segment-probabilities, and randomised decision trees, was proposed by Lasseck [37], which outperformed the other submitted methods in the LifeCLEF Bird task (BirdCLEF challenge) in 2014. Stowell et al. [38] studied an unsupervised feature learning method, which was demonstrated to be efficient in the

BirdCLEF 2014 challenge. Noticeably, the state-of-the-art *deep learning* methods were increasingly demonstrated to be excellent in performing the classification of large scale bird sound data. Promising results by using deep architectures for bird sound classification had been reported in recent BirdCLEF challenges [39–41, 242–244].

Generally, most existing studies on bird sound classification are focused on finding efficient features and classifiers whereas few studies focus on reducing the human expert annotation for unlabelled bird sound data (segmented syllables, or continuous recordings). Specifically, within the study of bird sound, there are large amounts of unlabelled audio recordings made in the field by ornithologists and amateurs, which bring forth a huge challenge for human annotators. Nevertheless, human annotation is time-consuming, expensive, and undesirable. To overcome the challenges mentioned above, *active learning* (AL) [204] is significant for the domain of bird sound classification. A pilot work was firstly proposed in [245], in which two basic AL algorithms modified from [83] were investigated and compared, i. e., *sparse-instance-based* AL (SI-AL), and *least-confidence-score-based* AL (LCS-AL). This preliminary work reported that for classifying 60 species or birds, using the AL paradigm via an SVM classifier can reduce up to 35.2% human annotations compared with randomly selecting samples. In addition, a previous work demonstrated that introducing *extreme learning machines* (ELMs) [194] can benefit the bird sound classification [246]. Moreover, it was also reported that combining the ELM-based AL paradigm can be superior to or at least comparable to SVM [247–249]. Motivated by the success of the aforementioned relevant work, a *kernel-based extreme learning machine* (KELM) [200] is introduced to the paradigm of AL. Two popular AL algorithms, SI-AL (cf. Section 2.5.2) and LCS-AL (cf. Section 7) are investigated and compared when using KELM and SVM. Furthermore, a detailed comparison on effectiveness and robustness of the algorithms will be given in this thesis. The main part of this study was previously published in [112].

3.2.2 Museum für Naturkunde Berlin Bird Sound Database

In this study, a publicly accessible bird sound database is provided from the Museum für Naturkunde Berlin (MNB), Berlin, Germany⁶. There are three main reasons for selecting this database: First, it can be accessed via a web-based interface, which makes the proposed study reproducible and sustainable. Second, compared with other popular public bird sound databases, e. g., the LifeCLEF Bird task [241], the MNB bird sound database only includes recordings that have a high acoustic quality. Furthermore, there is a rigid process of determination of species along with the location of the recording. Therefore, this database can be suitable to evaluate the algorithms, which aim for reducing the human annotations rather than only

⁶<http://www.animalsoundarchive.org/RefSys/Statistics.php>

Table 3.14: Number of instances and percentage of the total database for the initial training set, pool set and validation set, respectively.

Initial	Pool	Validation	Σ
539 (10 %)	3 478 (70 %)	1 043 (20 %)	5 060 (100 %)
1 030 (20 %)	2 987 (60 %)	1 043 (20 %)	5 060 (100 %)

recognising bird species by their sounds. Thirdly, the recordings included in this database were recorded at a variety of locations and within different equipment, which can make the study more robust and close to the real-world scenario.

The whole database originally contains 6 487 audio recordings from 273 species (subspecies) of birds. To guarantee an efficient initial training in the AL paradigm, the species which have less than 20 audio recordings were eliminated, which results in 86 species (5 060 audio recordings) for this study (refer to Table A.1 in Appendix A). The time duration of these recordings ranges from 0.3 s to 59.0 s within an average of 2.8 s. The total time duration of these recordings is approximately 4.0 hours. Among each species of bird sound, 20 % ($\approx 1 043$) of the instances were randomly selected for the validation set (see Table 3.14). In this study, there are two experiments that will be implemented, i. e., evaluation of the efficiency of the algorithms proposed in Section 2.5.2, and comparison of the algorithms' robustness. To fulfil the first experiment, an initial training set with a size of 10 % (≈ 539) from each species of bird sound was set up. In the second experiment, both of the two initial training sizes, i. e., 10 % (≈ 539), and 20 % ($\approx 1 030$) from each species of bird sound were used. In each experiment, the rest of the data will be the pool data set (unlabelled instances). The real labels of the pool data were fed to the classifiers as imitating the *human annotation* process.

3.2.3 Experimental Setup

Acoustic Features

In this study, the large scale acoustic feature set COMPARE (cf. Section 2.1.5) is chosen for its excellent performance in a previous relevant study [245]. This feature set is extracted by the OPENSIMILE toolkit [109, 110], which totally contains 6 373 features. Before feeding into the classifier, all the features are normalised (cf. Section 2.3) to a scale from 0 to 1. The parameters of the normalisation are calculated from an initial training set, and applied to the pool and validation sets.

Classification Models

Both for SVM and KELM, a *polynomial* kernel [250], $K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i^T \mathbf{x}_j + c)^{\hat{d}}$, is used for its excellent performance in initial experiments. In particular, the parameter γ is set to $1/6373$ and 1 for SVM and KELM, respectively. Both for SVM and KELM, c and \hat{d} are set to 1 and 10 , respectively. These parameters are all based on initial empirical experiments. The C_s -value (cf. Section 2.4.1) and C_e -value (cf. Section 2.4.3) for SVM and KELM are set to be the same as 10 optimised by a searching grid of $\{10^{-5}, 10^{-4}, \dots, 10^4, 10^5\}$. SVM is implemented by the open source toolkit LIBSVM [211]. KELM is implemented by MATLAB scripts provided online⁷.

PL vs. AL

The performance of randomly selecting unlabelled instances for human annotation, i. e., *passive learning* (PL) is taken as the baseline to be compared with AL (SI-AL and LCS-AL). The implementations of the PL and AL algorithms are written in MATLAB scripts. The parameter λ_s (cf. Section 2.5.2) for SI-AL, and the parameter λ_c for LCS-AL (cf. Section 7) are set to 0.5 and 0.1 , respectively. UAR (cf. Section 2.7.1) is selected as the evaluation metric for considering the unbalanced character of bird sound data. To evaluate the effectiveness of AL (Section 3.2.4), an initial training size of 539 labelled instances is used. To compare the robustness of the algorithms, two initial training sizes, i. e., 539 and 1030 , are used. In the following experiments, all of the pool data set will be used during the final iteration of each algorithm to make a comprehensive comparison.

3.2.4 Comparison of Passive Learning and Active Learning

To evaluate the effectiveness of AL, an initial training set with a size of 539 labelled instances is fed into the classifier (refer to Table 3.14). The UARs achieved by the trained classifier through the whole AL iterative process versus the corresponding human labelled instances from the pool data set are shown in Figure 3.5. The experiments for PL are run independently 20 times to make a thorough fulfilment of PL due to its nature of randomly selecting unlabelled data for human annotation. It can be seen that, AL is more efficient in improving the trained classifier's performance than PL for both SVM and KELM. Particularly, LCS-AL (for both SVM and KELM) can reach the best recognition performance during its very early iterations. Even though by SVM, SI-AL cannot considerably enhance the classifier's performance as LCS-AL; it can still be comparable to the maximum UAR value reached by PL at each iteration step (see Figure 3.5(a)). For AL algorithms implemented by KELM, both SI-AL and LCS-AL outperformed PL in improving the

⁷<http://www.ntu.edu.sg/home/egbhuang/>

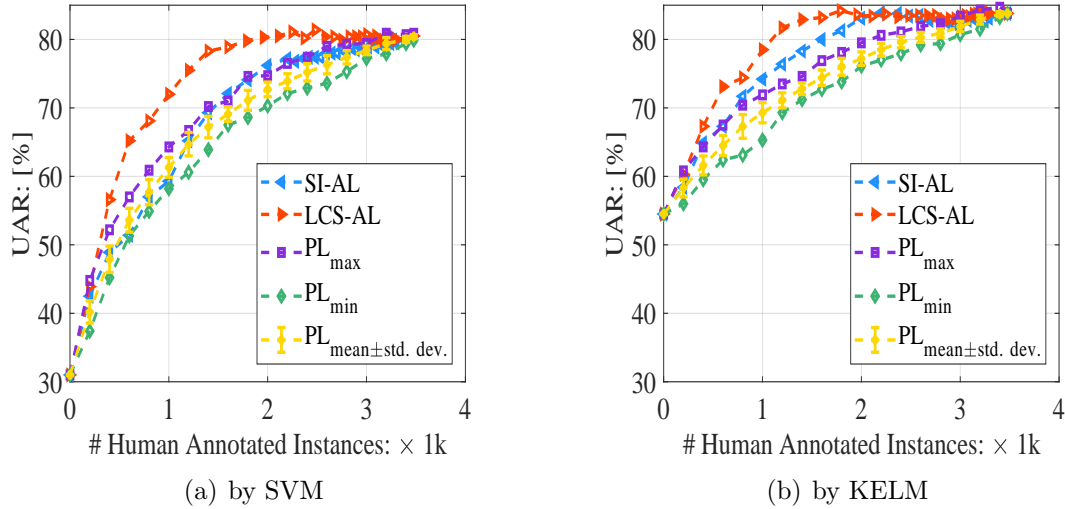


Figure 3.5: Comparison of UARs vs the number of human annotated instances between algorithms with 539 initial supervised training instances. The measures for PL are shown with 20 independent runs.

trained classifier’s performance when using the same number of human annotated instances.

Generally, the experiments above demonstrate that, when using the same number of human annotated instances, AL algorithms are superior to PL in improving the capacity of the trained classifier. In other words, AL can save more human annotation work than PL when the task needs an acceptable recognition performance. Furthermore, LCS-AL is found to be more efficient than SI-AL in selecting the ‘most informative’ samples. Another phenomenon that should be noted is that, compared with PL, AL tends to have a slight decrease in improving the classifier’s performance after reaching its highest point. This can be explained as when the ‘most informative’ samples are being fed into the classifier, other samples will bring uncertain information to the model.

3.2.5 Comparison of Robustness

To make a comparison of the robustness of the proposed algorithms, in this section, two scales were selected for the initial training set, i. e., 539 and 1030 (refer to Table 3.14). Meanwhile, the performances of the two classifiers, i. e., SVM and KELM, will be investigated in this study. For each comparison group, the two scales mentioned above are randomly generated and equally fed into different algorithms by 20 independent runs. It should be noted that in the experiments, the algorithms had different iteration steps. More specifically, for SI-AL, it normally needed longer

time to use up all the pool data set than for LCS-AL and PL. Therefore, the results (UARs) are shown only in common learning iterations of PL, SI-AL, and LCS-AL, respectively. In addition, considering the difference of the initial performance by SVM and KELM (see Figure 3.5), a common UAR's range of improvement (from 60.0% to be 80.0%) is evaluated for measuring the two classifiers' capacity to reduce human annotation work when applied to the AL paradigm. As a baseline, experimental results on PL algorithm will also be presented.

The mean and standard deviations of UARs averaged across 20 independent runs of each algorithm versus human annotated instances are shown in Figure 3.6. It can be seen that, for both SVM and KELM, LCS-AL is the fastest algorithm to improve the classifier's performance at early iterations. All the standard deviations of UARs exhibit a descending trend, which indicates that the classifier can be more stable when adding more human annotated instances. Particularly, compared with PL, LCS-AL can improve the classifier's performance using much less human annotated instances, and with less instability (smaller standard deviations than PL). In this study, SI-AL is not demonstrated to be more efficient and robust in reducing the human annotated instances than PL, especially by SVM.

Figure 3.7 illustrates the percentage (in statistical box plots) of the used human annotated instances in the pool data set by various algorithms within the UAR's improvement ranging from 60.0% to 80.0%. As consistent with the findings in Figure 3.6, for both SVM and KEML, LCS-AL can be superior to SI-AL in improving the classifier's capacity using less human annotations. Comparing classifiers, KELM shows more strengths than SVM in reducing human annotated instances when applied to the AL paradigm. In particular, SI-AL can fulfil its function of reducing human annotation work compared with PL when it uses KELM as the classifier. More details (the minimum, maximum, mean, and median values of percentage in [%]) from Figure 3.7 are listed in Table 3.15 (for an initial training size of 539) and Table 3.16 (for an initial training size of 1 030), respectively. For SVM, LCS-AL can reduce performance from approximately 20-29% (within 539 initial training instances) to more than approximately 30-35% (within 1 030 initial training instances) human annotated instances from pool data set compared with PL, while SI-AL might use slightly more human annotations than PL. In contrast, SI-AL works well for KELM, which can considerably reduce the human annotation work by PL (approximately 17-25% to 6-20%). Among the various algorithms, LCS-AL within the use of KELM occupies the best place in this study by reducing from approximately 35-40% to approximately 33-47% human annotated instances compared to PL.

It should be noted that, in Figure 3.6, given a larger size of the initial training set i. e., 1 030 human annotated instances, the models (for both SVM and KELM) can have smaller deviation at the beginning. However, most of the AL algorithms' performances on reducing human annotation work have a slight decrease (see Table 3.15 and Table 3.16). This phenomenon could be explained as when adding

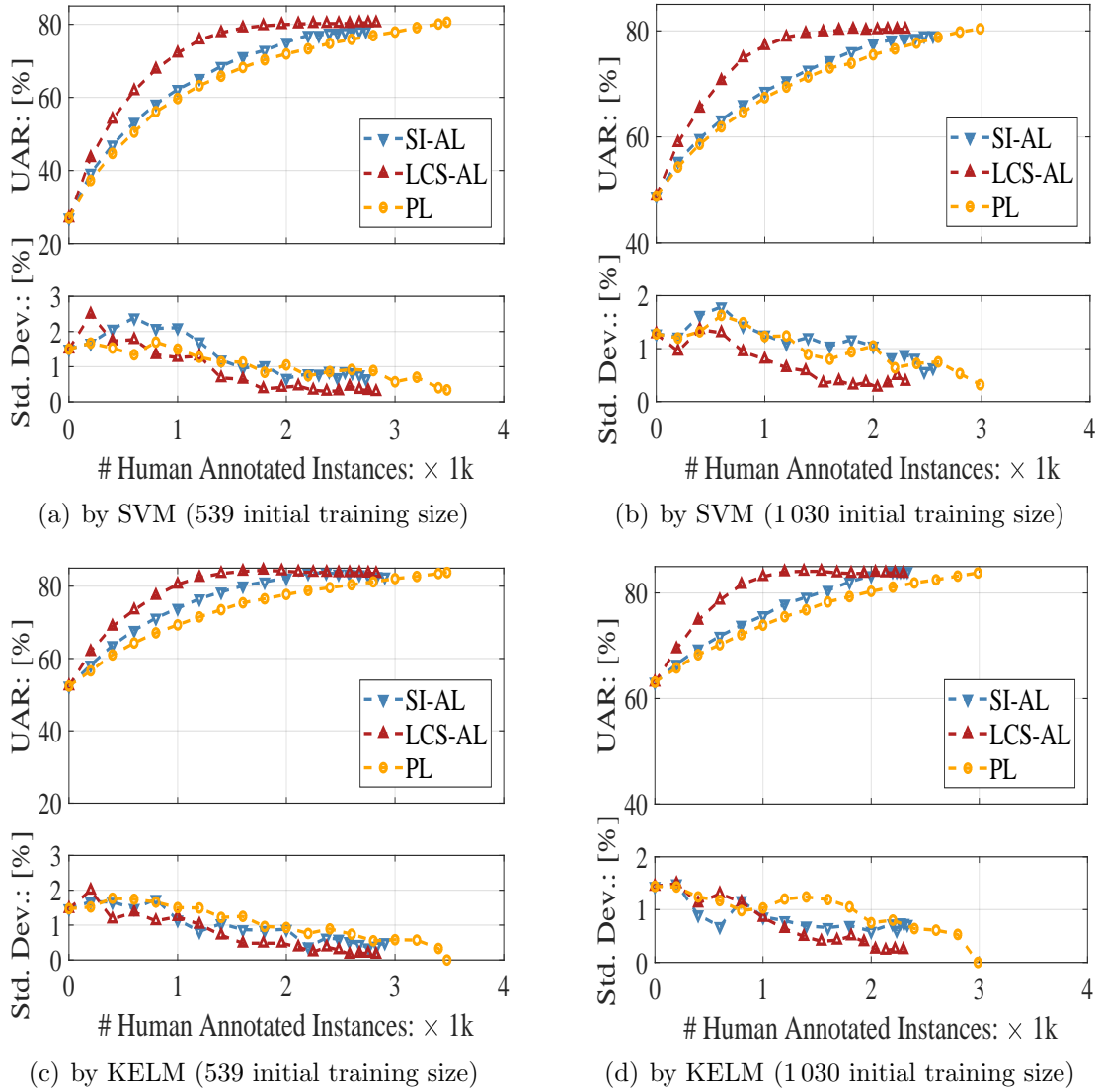


Figure 3.6: Comparison of UARs vs the number of human annotated instances between algorithms across 20 independent runs (both for the averaged UAR and standard deviation). The charts only illustrate the UARs in common iterations of PL, SI-AL, and LCS-AL, respectively.

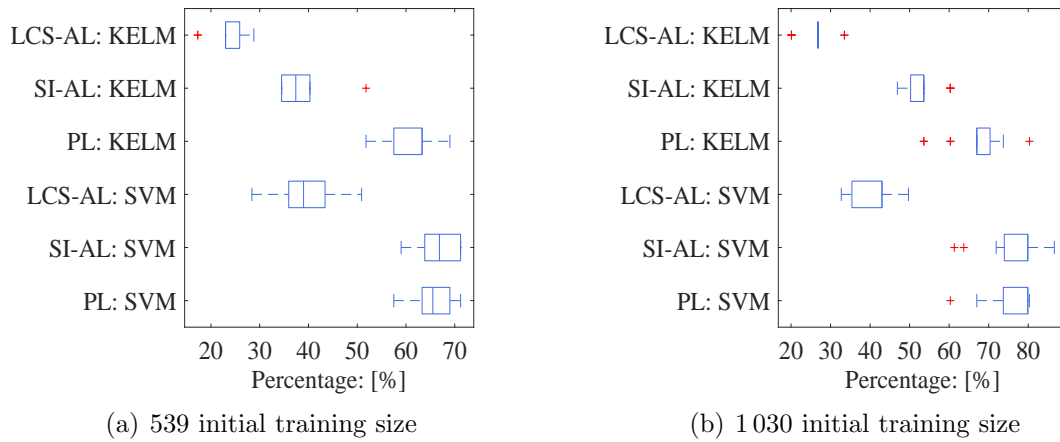


Figure 3.7: Boxplots of the percentage of used human annotated instances in the total pool data when the performance (UAR) was improved from 60.0 % to 80.0 %.

more initial *human* annotated instances, the initial classifier could build more stable (smaller deviation at beginning). On the other hand, more unavoidable mistakes by *human* experts might be involved when using a larger *human* annotated initial training set, which could lead to a worse performance of the classifier to select the ‘most informative’ data in the following learning iterations (a slight decrease on AL’s performance).

In order to explore more details, one-tailed Student’s *t*-test [213] is used to make a statistical analysis of UARs across iterations by various algorithms (see Table 3.17 and Table 3.18). To eliminate the effect of early iterations’ instability and to meanwhile guarantee a common length for each algorithm, the one-tailed Student’s *t*-test was set from the 4-th to the 18-th iteration by each algorithm within 539 initial training instances, and the 4-th to the 15-th iteration by each algorithm within 1 030 initial training instances. The comparisons are made between a pair of various algorithms listed in the left column and the top row of each table. The significance levels are illustrated by grayscale shading, based on the values $p < .05$, $p < .01$, and $p < .001$. The analysis of both tables confirms the previous observations and indicates that, among the algorithms, LCS-AL is significantly better than SI-AL and PL. For classifiers, KELM can significantly outperform SVM when applied to the same AL algorithm, e. g., SI-AL or LCS-AL. The LCS-AL algorithm using KELM as the classifier shows its best performance and robustness in this study.

3.2.6 Summary

Bird sound, as a typical and prevalent animal sound, has been studied for decades. Recognition of bird species by their sounds has been continuously attracting ex-

Table 3.15: The percentage [%] of used human annotated instances when the performance (UAR) was increased from 60.0% to 80.0% with 539 initial supervised training instances.

		Min	Max	Mean	Median
SVM	PL	57.5	71.2	65.8	65.5
	SI-AL	59.0	71.2	66.8	66.9
	LCS-AL	28.4	50.9	40.0	39.0
KELM	PL	51.8	69.0	61.5	63.3
	SI-AL	34.5	51.8	38.0	37.4
	LCS-AL	17.3	28.8	23.9	23.0

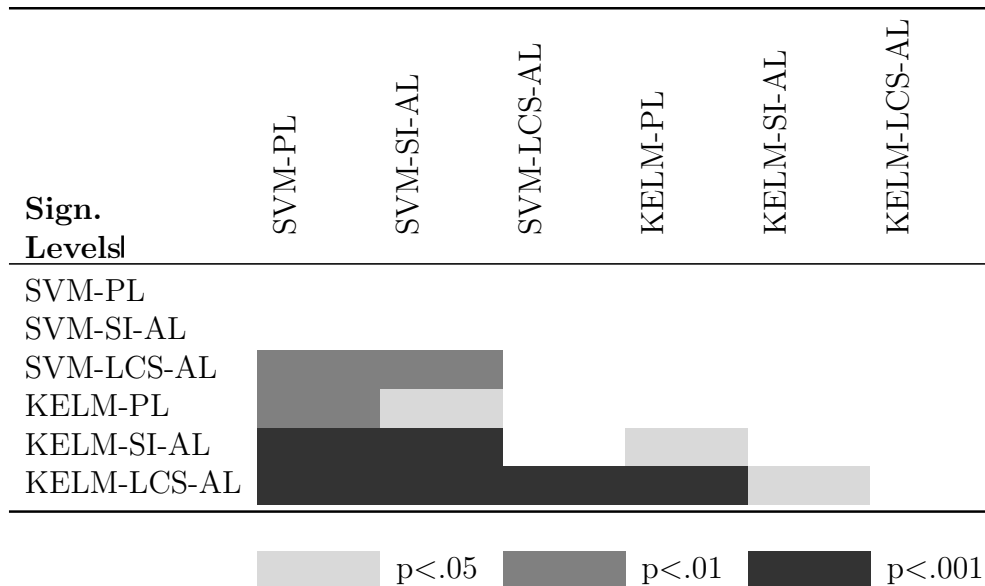
Table 3.16: The percentage [%] of used human annotated instances when the performance (UAR) was increased from 60.0% to 80.0% with 1030 initial supervised training instances.

		Min	Max	Mean	Median
SVM	PL	60.3	80.3	76.7	79.9
	SI-AL	61.3	86.6	77.6	79.9
	LCS-AL	32.7	49.7	40.6	43.0
KELM	PL	53.6	80.3	67.0	67.0
	SI-AL	46.9	60.3	53.5	53.6
	LCS-AL	20.1	33.5	26.1	26.8

perts from the research community of *ecology*, *bioacoustics*, *signal processing*, and *machine learning* to work together to develop a long-term, non-human acoustical monitoring system for measuring the activities of birds, which can be an important fingerprint of the state of nature [22], climate change [23], and biodiversity in local ecosystem [24, 25]. Nevertheless, an important fact had been ignored in previous bird sound classification study, namely that, there are large amount of bird sound data in the real-world that are not annotated by human experts. Besides, it would be expensive or even difficult for human experts to annotate the large amount unlabelled bird sound data. In this section, *active learning*, had been investigated for reducing the human annotation work for bird sound recordings.

Firstly, the effectiveness of AL algorithms had been demonstrated in Section 3.2.4. Compared with PL (randomly selecting unlabelled bird sound data for human annotation), AL (selecting the ‘most informative’ unlabelled bird sound data for human annotation) can truly reduce the human annotation work. Particularly,

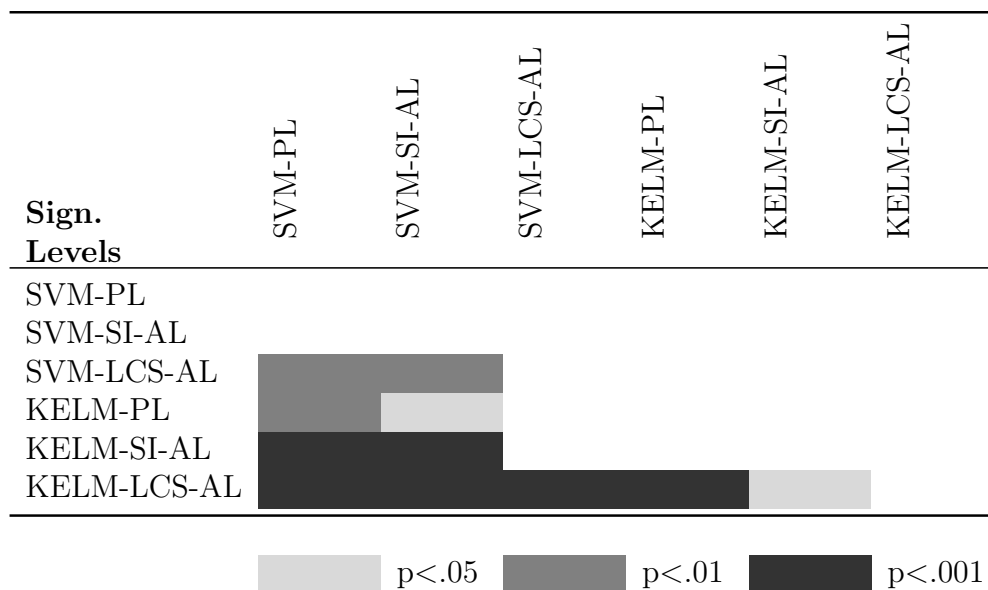
Table 3.17: Significance levels of the averaged UARs obtained from the statistical comparison (one-tailed Student’s t -test) between iteration: 4 to 18 with 539 initial supervised training instances (800 ~ 3 478 instances for SVM-PL, 200 ~ 2 732 (averaged) instances for SVM-SI-AL, 800 ~ 2 674 (averaged) instances for SVM-LCS-AL, 800 ~ 3 478 instances for KELM-PL, 200 ~ 2 906 (averaged) instances for KELM-SI-AL, 800 ~ 2 927 (averaged) instances for KELM-LCS-AL.)



for SVM, SI-AL cannot considerably reduce the human annotation work while it can be comparable to the maximum performance of PL’s 20 independent runs of experiments in Section 3.2.4.

Secondly, a comparison on robustness of various algorithms was presented in Section 3.2.5. In contrast to Section 3.2.4, there were two scales of initial training sets, i. e., 539 and 1 030 instances, fed into the classifier. It was found that, for SVM, SI-AL did not show any improvement compared with PL. However, LCS-AL can perform significantly better than PL when using SVM as the classifier (refer to Table 3.17 and Table 3.18). This might be the reason that LCS-AL can be well-matched to SVM’s boundary-learning behaviour [81]. In this study, a ‘sampling margin value’ (MSV) was used in LCS-AL as the measure of the *confidence score*, which can be excellent for distinguishing the two most similar possible classes of a given sample. In addition, KELM’s superior performance to SVM had been demonstrated through the whole study in this section. This could be explained mainly by two reasons: First, when a same kernel (e. g., *polynomial kernel* in this study) was used, SVM tends to find a solution sub-optimal to KELM’s solution [199]. Second, KELM can be directly applied to multi-class cases while SVM has to convert

Table 3.18: Significance levels of the averaged UARs obtained from the statistical comparison (one-tailed Student's t -test) between iteration: 4 to 15 with 1 030 initial supervised training instances (800 ~ 2 987 instances for SVM-PL, 200 ~ 2 552 (averaged) instances for SVM-SI-AL, 800 ~ 2 550 (averaged) instances for SVM-LCS-AL, 800 ~ 2 987 instances for KELM-PL, 200 ~ 2 334 (averaged) instances for KELM-SI-AL, 800 ~ 2 287 (averaged) instances for KELM-LCS-AL.)



and indirectly solve the multi-class problems to some type of binary classification problems, which might change the application property and distribution [199].

3.3 Robust Systems for Acoustic Scene Classification

In this study, the wavelet features, WPTE (cf. Section 2.1.6) and WEF (cf. Section 2.1.6) are introduced to the application of acoustic scene classification. When combining these with the large scale feature set COMPARE, wavelet features can contribute to an improvement of the model’s performance, specifically in noisy environments. Section 3.3.1 firstly gives a background on the study of acoustic scene classification. In the following, a publicly accessible database will be introduced in Section 3.3.2. Then, the experimental setup will be described in Section 3.3.3. The experiments are done in clean and noisy environments, which are discussed in Section 3.3.4 and Section 3.3.5, respectively. Finally, Section 3.3.6 summarises this work and gives possible future directions.

3.3.1 Background

Acoustic scene classification (ASC) is a subfield of *computational auditory scene analysis* (CASA) [47], which has been studied for more than two decades [57]. ASC refers to the task of predicting an audio stream where the audio was recorded (e. g., *beach* or *park*). More precisely, in this thesis, ASC refers to developing computational algorithms that can automatically perform audio classification using signal processing and machine learning (refer to [57]). Relevant applications of ASC can benefit areas like multimedia searching [50], smart mobile devices [51], intelligent monitoring systems [52, 53], and public/home security surveillance [54–56]. A literature survey summarises the early work on features and classifiers for the ASC task [57]. The popularly used features include *Mel-frequency cepstral coefficients* (MFCCs), *linear predictive coefficients* (LPCs), etc. As to classifiers, *hidden Markov models* (HMMs), *Gaussian mixture models* (GMMs), and *support vector machines* (SVMs) were used. A detailed introduction and comparison of the features and classifiers aforementioned can be found in [57]. Recently, advanced technologies in *deep learning* have become a mainstream in ASC [64–77]. In particular, using *convolutional neural networks* (CNNs) [160, 162, 236] to extract higher representations from the spectrum of audio scene recording can save time on designing human hand-crafted features and usually outperforms the conventional acoustic features.

Generally, most investigated features (including the CNN-based representations) are based on *Fourier* transformation [78]. The Heisenberg-alike time-frequency trade-off [79] restrained the Fourier-based short time analysis of the signal to not have a good time and frequency resolution at the same time. Unlike Fourier transformation, the *wavelet transformation* (WT) can be applicable to reach a multi-resolution of the signal [84, 119]. However, the existing studies on introducing wavelet features into the ASC task are limited. Rabaoui et al. used a combination of wavelet

decomposition features and other widely used features (e., g., MFCCs) to train several one-class SVM classifiers to classify nine classes of acoustic scenes [55], in which the final results could reach an accuracy of approximately 97.0%. Li et al. combined wavelet and MFCC features to feed a treebagger classifier, which can achieve an accuracy of 72.0% for classifying 10 acoustic scenes [251]. The main drawback of these studies is that they are focused on a very small size database (instance number < 1500). In addition, in the work by Rabaoui et al. [55], the model trained individually by wavelet features was not strong enough, which might need more advanced wavelet-based features to be compared against.

In this thesis, two kinds of wavelet features, WPTE (cf. Section 2.1.6) and WEF (cf. Section 2.1.6) are applied to the ASC task. In addition, the large scale feature set COMPARE (cf. Section 2.1.5) will also be investigated and combined with the aforementioned wavelet features within a late fusion (cf. Section 2.6). As to classifiers, an SVM classifier (cf. Section 2.4.1) and deep learning models (GRNN and BGRNN, cf. Section 2.4.2) will be used. The evaluation will be measured in both clean and noisy environments.

3.3.2 DCASE 2017 Acoustic Scene Database

The released database, i. e., the TUT Acoustic Scenes 2017 (for task 1: acoustic scene classification) of the IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE 2017) [48] can be accessed via the challenge website⁸. The whole database contains 312 and 108 segments of 10 seconds in each of 15 classes in the *development* and *test* (evaluation) set, respectively. The total duration for the development and test set is 13.0 and 4.5 hours, respectively. The fifteen acoustic scene classes needed to be recognised include: *beach*, *bus*, *cafe/restaurant*, *car*, *city centre*, *forest path*, *grocery store*, *home*, *library*, *metro station*, *office*, *park*, *residential area*, *train*, and *tram*. Compared with the previously released database [86], in this edition of the challenge, the *instances* are shorter (10 second segments) in length (30 seconds in [86]). This length of recordings, i. e., 10 seconds, provides less information to the system for decision making process. It can be challenging for both human and machine recognition [51]. In addition, in this edition of the challenge, the *test* set was newly recorded, which needs a high robustness of the trained system. In the phase of the development, the data was independently (based on the origination of recordings) split into four folds to conduct a cross-validation for developing the system. The evaluation of the development set will be based on the averaged *accuracy* of the system achieved among the four-fold cross-validation. In the phase of the test, as was done in the official baseline [48], all the data from the development set will be used to train the system when evaluating the test set. The performance will also be measured by the metrics of accuracy.

⁸<http://www.cs.tut.fi/sgn/arg/dcase2017/>

Table 3.19: The parameters for extracting wavelet features for the task of acoustic scene classification. J_{max} : maximum decomposition level. Dim: dimension.

	Wavelet Type	J_{max}	Dim of LLDs	Dim of Feature Set
WPTE	‘rbio3.3’	7	255	1 020
WEF	‘db7’	7	287	1 148

3.3.3 Experimental Setup

In this study, to make a direct comparison with the *baseline system* [48], *accuracy* will be used as the evaluation metrics. The results shown in the development are averaged values of four-fold cross-validation as was setup in the official baseline [48]. The test set will be used to evaluate the system validated and trained with all the data from the development set. The whole experimental setup is described as follows:

Baseline System

The official baseline system⁹ is implemented with log mel-band energies as the features and a multilayer perceptron (MLP) as the classifier. The features are calculated within a frame of 40 ms (20 ms as the overlap) to extract 40 mel bands covering the frequency range from 0 to 22.05 kHz. To construct the feature vector, a 5-frame context is used resulting in a length of 200. The MLP consists of two-layers of 50 hidden units each (*drop out rate*: 20%, *epoch*: 200, *learning rate*: 0.001). The output layer of the network consists of *softmax* type neurons representing 15 classes, which can be active only one at a time. The prediction will be based on the *majority voting* for frame-based decisions to obtain a single label per classified segment. The system is based on Keras¹⁰, on which more detailed information can be found [48].

Proposed Systems

The proposed systems use COMPARE (cf. Section 2.1.5), WPTE (cf. Section 2.1.6), and WEF (cf. Section 2.1.6) feature sets (applied within *functionals*). On the back-end side, the SVM (cf. Section 2.4.1), GRNN (cf. Section 2.4.2), and BGRNN (cf. Section 2.4.2) are used. The parameters for both features and classifiers were optimised in initial experiments on the development set and applied to the test set. The frame sizes and overlaps for extracting LLDs of the aforementioned three feature sets are all set to be consistent to the configuration in the baseline system, i. e., 40 ms

⁹<https://github.com/TUT-ARG/DCASE2017-baseline-system>

¹⁰<https://keras.io/>

frame with 20 ms overlap. The wavelet type and the maximum decomposition level J_{max} are listed in Table 3.19. The names of the wavelet types and the decomposition scripts are based on the Wavelet Toolbox of MATLAB by MathWorks as mentioned in Section 3.1.3. Functionals used for ComParE, WPTE, and WEF are the same as used in Section 3.1.3. The SVM classifier is implemented by the popular toolkit LIB-SVM [211] with a *linear* kernel and the C_s -value is set to 0.1. The GRNN/BGRNN classifier is implemented by Python scripts based on TensorFlow¹¹ and TFLearn¹² with an architecture of 120 and 160 GRU hidden units respectively in the first and second layer (*drop out rate*: 10%, *epoch*: 50, *learning rate*: 0.0002). The optimisation method is set to RMSprop [252]. More specifically, when extracting the features (in the format of functionals) fed into the GRNN/BGRNN classifier, the original instance (10 second segment) was cut to *episodes* (of 1 second) sequenced by time steps of 0.5 seconds. In the phase of late fusion (cf. Section 2.6), there are two strategies that will be used and compared, i. e., *majority voting* (MV, can be referred to Section 2.6), and *margin sampling voting* (MSV, can be referred to Section 2.6). The decision will be made by adopting the fusion strategies from the predictions made by the classifier trained independently with various feature sets.

Noisy Environments

In this study, white Gaussian noise was added to the original audio recordings to evaluate the system’s robustness in noisy environments. There are five levels of the signal-to-noise ratio (SNR) at -10 dB, -5 dB, 0 dB, 5 dB, and 10 dB, respectively. It should be noted that, even though there are numerous advanced signal processing techniques that can fulfil the task on noise reduction [253], the main direction of this study is to investigate the effectiveness of the proposed systems in low SNR environments rather than signal enhancement at the front-end side.

3.3.4 Acoustic Scene Classification in Clean Environments

The accuracies achieved by each kind of classifier when feeding with different feature sets in a clean environment (noise-free) are listed in Table 3.20. The results on the development set are best performances of models trained within optimised parameters, which are applied to the model for test set. It can be seen that, over half of the whole results can outperform the official baseline. Most of these performances are achieved by the fused models using the strategy of MV (*majority voting*, cf. Section 2.6) or MSV (*margin sampling voting*, cf. Section 2.6). The three best models that show significance levels ($p < .01$, $p < .05$, and $p < .05$, respectively by one-tailed z -test) are all based on the SVM classifier.

¹¹<https://github.com/tensorflow/tensorflow>

¹²<https://github.com/tflearn>

3. Applications

Table 3.20: Results (accuracies in [%]) achieved by various classifiers when feeding within different feature sets in clean environment. The official baseline of DCASE 2017 Challenge Task 1 is 74.8 for development set, and 61.0 for test set (can be referred to [48]). Bold entries represent the results better than the official baseline (on test set). Results (on test set) showing significance levels are highlighted by grayscale shading, based on the values $p < .05$, and $p < .01$ by one-tailed z-test. Dev: Development.

		SVM		GRNN		BGRNN	
		Dev	Test	Dev	Test	Dev	Test
COMPARÉ		77.9	61.3	78.3	58.6	75.8	59.0
WPTE		75.5	58.7	73.7	55.7	73.2	54.6
WEF		77.8	60.4	77.1	63.3	75.7	60.3
COMPARÉ+WPTE	MV	76.4	60.6	76.0	57.3	65.6	60.1
	MSV	82.1	65.0	81.0	60.7	65.6	61.9
COMPARÉ+WEF	MV	76.9	63.1	77.3	63.3	75.6	63.6
	MSV	82.9	63.6	82.7	63.0	81.4	62.5
WPTE+WEF	MV	76.6	60.4	75.1	59.5	74.9	58.3
	MSV	78.7	61.2	77.1	61.7	76.1	59.6
ALL	MV	80.9	63.8	80.1	62.5	78.5	63.4
	MSV	83.1	64.6	82.5	63.0	81.1	62.2

$p < .05$
 $p < .01$

As consistent with most of the submissions to DCASE 2017 Task 1¹³, there are big gaps between the results on the development set and the test set. It could be explained by the fact that, the test set was recorded newly compared with the development set [48], which could lead the trained models on the development set to overfit.

In this clean environment, the COMPARÉ feature set shows comparable performances (e. g., 61.3% by SVM classifier) to wavelet features (e. g., WEF reaches 60.4% by SVM classifier). However, it still yields to the models fused with wavelets. In addition, when comparing the performance of the classifier trained by a single feature set, WEF with a GRNN classifier can reach the best place (63.3%). These results support the previous findings in [113] that, introducing wavelet features can

¹³<http://www.cs.tut.fi/sgn/arg/dcase2017/challenge/task-acoustic-scene-classification-results>

considerably benefit the model’s performance on acoustic scene classification. Moreover, as consistent with the results in [113], GRNN cannot be superior to SVM in this study. The reason might be explained by the reason that, the recordings are short time duration (10 seconds), which restrains the capacity of GRNN to learn sufficient context information of a certain acoustic scene. Additionally, unlike human speech conversation, learning the information from past and future states could not better the predictions by BGRNN. Therefore, in this study, the performance of BGRNN does not show significant improvement compared with GRNN.

3.3.5 Acoustic Scene Classification in Noisy Environments

The results of the proposed systems in noisy environments (SNR level at -10 dB, -5 dB, 0 dB, 5 dB, and 10 dB, respectively) are shown in Table 3.21 (by an SVM classifier), Table 3.22 (by a GRNN classifier), and Table 3.23 (by a BGRNN classifier), respectively. In noisy environments, the superiority of proposed systems to official baseline system is more significant than that in the clean environment. More precisely, in low SNR environment (e. g., -10 dB), most of the fused models (using MSV strategy) can outperform the baseline system at a significance level of $p < .001$ by one-tailed z -test. Both COMPARE and wavelet features have shown their robustness in noisy environments. As consistent with the findings in the clean environment, introducing wavelets by a late fusion strategy can contribute to an improvement of the system. Generally, the performances of GRNN/BGRNN are only comparable to SVM, which might be due to the same reason as in the clean environment, i. e., there is limited context information in acoustic scenes.

To find more details on the performance of the baseline system and the best proposed system in clean and noisy environments, the class-wise accuracies on the test set are illustrated in Table 3.24. In the clean environment, the top three class-wise accuracies improved by the best propose system compared with the baseline system are *cafe/restaurant* (from 43.5 % to 74.1 %, $p < .001$ by one-tailed z -test), *office* (from 73.1 % to 91.7 %, $p < .001$ by one-tailed z -test), and *beach* (from 40.7 % to 57.4 %, $p < .01$ by one-tailed z -test), respectively. The counterparts in the low SNR (-10 dB) environment are *car* (from 0.0 % to 29.6 %, $p < .001$ by one-tailed z -test), *forest path* (from 43.5 % to 70.4 %, $p < .001$ by one-tailed z -test), and *residential area* (from 3.7 % to 27.8 %, $p < .001$ by one-tailed z -test), respectively. Nevertheless, there are some acoustic scenes having a decrease in the best proposed system, e.g., *park* in the low SNR (-10 dB) environment (from 28.7 % to 15.7 %, $p < .01$ by one-tailed z -test).

3.3.6 Summary

In this study, three feature sets, i. e., COMPARE (cf. Section 2.1.5), WPTE (cf. Section 2.1.6), and WEF (cf. Section 2.1.6) were evaluated for the task of acoustic scene

3. Applications




Table 3.21: Results (accuracies in [%]) achieved by an SVM classifier when fed with different feature sets in noisy environments. The results (on the test set) by the official baseline system [48] are: 32.8 (-10 dB), 40.2 (-5 dB), 39.9 (0 dB), 45.9 (5 dB), and 49.6 (10 dB). The bold entries represent the results that are better than the official baseline system (on the test set). The results (on the test set) showing significance are highlighted by grayscale shading, based on the values $p < .05$, $p < .01$ and $p < .001$ by one-tailed z-test. Dev: Development. SNR [dB]: Signal-to-Noise Ratio. Dev: development.

SNR: dB			-10	-5	0	5	10
COMPARÉ	Dev		55.4	60.4	64.4	70.0	71.9
	Test		39.0	45.2	49.0	52.7	54.3
WPTE	Dev		45.7	53.0	56.6	60.2	63.2
	Test		34.4	36.9	41.7	46.1	49.3
WEF	Dev		52.5	55.7	60.8	64.5	66.8
	Test		39.3	43.3	44.8	45.5	48.0
COMPARÉ+WPTE	MV	Dev	51.0	57.7	61.9	66.5	68.6
		Test	36.4	38.9	43.8	48.3	52.0
	MSV	Dev	56.4	61.2	64.9	69.5	72.4
		Test	39.6	44.8	49.3	52.7	54.6
COMPARÉ+WEF	MV	Dev	54.9	59.7	63.8	67.8	70.0
		Test	39.0	42.9	46.7	49.6	52.2
	MSV	Dev	58.6	62.8	66.2	70.8	73.2
		Test	40.9	47.2	50.1	51.8	53.8
WPTE+WEF	MV	Dev	49.1	54.7	59.2	62.7	65.6
		Test	36.2	39.4	42.9	45.4	49.3
	MSV	Dev	52.3	56.4	60.2	64.4	67.5
		Test	39.9	42.2	45.4	46.9	49.1
ALL	MV	Dev	55.3	59.9	64.2	67.8	70.4
		Test	41.0	44.3	46.9	48.9	53.3
	MSV	Dev	58.1	62.1	65.4	70.1	73.0
		Test	41.9	46.2	49.9	52.1	53.4

 $p < .05$
 $p < .01$
 $p < .001$

Table 3.22: Results (accuracies in [%]) achieved by a GRNN classifier when fed with different feature sets in noisy environments. The results (on the test set) by the official baseline system [48] are: 32.8 (-10 dB), 40.2 (-5 dB), 39.9 (0 dB), 45.9 (5 dB), and 49.6 (10 dB). The bold entries represent the results that are better than the official baseline system (on the test set). The results (on the test set) showing significance are highlighted by grayscale shading, based on the values $p < .05$, $p < .01$ and $p < .001$ by one-tailed z-test. Dev: Development. SNR [dB]: Signal-to-Noise Ratio. Dev: development.

SNR: dB			-10	-5	0	5	10
COMpARE	Dev		51.4	59.9	63.5	66.2	70.1
	Test		37.0	42.4	49.7	53.6	56.7
WPTE	Dev		32.6	39.2	47.7	53.8	57.5
	Test		23.8	31.0	38.0	38.8	45.1
WEF	Dev		51.7	57.1	60.7	62.8	63.5
	Test		39.4	42.1	40.8	44.4	44.9
COMpARE+WPTE	MV	Dev	40.4	47.6	55.7	59.2	65.8
		Test	29.9	42.0	39.2	47.0	50.4
	MSV	Dev	51.0	59.3	63.1	66.7	70.3
		Test	36.4	41.5	49.8	54.0	55.9
COMpARE+WEF	MV	Dev	53.2	59.3	62.9	64.9	67.6
		Test	35.0	41.4	42.7	48.8	49.1
	MSV	Dev	53.4	61.8	65.2	68.5	71.7
		Test	39.8	44.6	49.1	54.0	56.5
WPTE+WEF	MV	Dev	41.4	46.7	55.8	57.9	62.6
		Test	32.5	39.4	37.8	42.6	44.1
	MSV	Dev	50.2	55.8	58.9	62.7	63.6
		Test	39.8	40.4	41.5	43.6	46.4
ALL	MV	Dev	49.4	56.8	61.8	65.0	67.9
		Test	33.8	41.7	45.2	47.3	51.9
	MSV	Dev	53.1	61.3	64.9	68.4	71.4
		Test	39.8	44.1	49.1	54.1	56.0

 $p < .05$  $p < .01$  $p < .001$

3. Applications

Table 3.23: Results (accuracies in [%]) achieved by a BGRNN classifier when fed with different feature sets in noisy environments. The results (on the test set) by the official baseline system [48] are: 32.8 (-10 dB), 40.2 (-5 dB), 39.9 (0 dB), 45.9 (5 dB), and 49.6 (10 dB). The bold entries represent the results that are better than the official baseline system (on the test set). The results (on the test set) showing significance are highlighted by grayscale shading, based on the values $p < .05$, $p < .01$ and $p < .001$ by one-tailed z-test. Dev: Development. SNR [dB]: Signal-to-Noise Ratio. Dev: development.

SNR: dB			-10	-5	0	5	10
COMPARe	Dev		51.2	57.2	61.0	65.0	69.6
	Test		36.3	41.7	45.7	55.2	53.4
WPTE	Dev		31.9	41.8	49.5	55.2	55.9
	Test		23.6	32.7	37.5	44.7	46.2
WEF	Dev		51.2	58.5	59.3	61.5	62.4
	Test		36.7	42.6	42.5	45.7	49.9
COMPARe+WPTE	MV	Dev	39.3	50.2	55.8	59.8	65.4
		Test	24.5	33.7	44.9	48.3	50.7
	MSV	Dev	50.6	57.2	61.1	66.1	70.2
		Test	36.1	41.3	45.3	53.7	53.2
COMPARe+WEF	MV	Dev	51.5	58.2	60.7	62.8	66.1
		Test	36.5	41.0	43.3	52.3	52.0
	MSV	Dev	54.8	61.1	63.3	67.5	70.9
		Test	39.6	42.7	47.3	52.4	53.6
WPTE+WEF	MV	Dev	39.0	50.5	54.9	58.4	61.3
		Test	29.0	34.7	40.1	45.0	48.5
	MSV	Dev	48.9	57.3	58.1	61.8	62.4
		Test	35.7	41.2	39.9	46.2	48.6
ALL	MV	Dev	48.2	56.9	61.0	64.6	67.6
		Test	34.3	42.5	44.8	49.4	53.0
	MSV	Dev	54.4	60.7	63.0	67.7	70.6
		Test	39.4	42.7	46.0	51.9	53.3

 $p < .05$
 $p < .01$
 $p < .001$

Table 3.24: Class-wise accuracies (in [%]) on test set of the baseline system and the best proposed system (highlighted by bold entries) in clean and low SNR (-10 dB) environment. The best proposed system in clean environment: a MSV fusion of COMPARE and WPTE by SVM classifier. The best proposed system in low SNR (-10 dB) environment: a MSV fusion of COMPARE, WPTE and WEF by SVM classifier.

Acoustic Scene	clean	-10 dB
<i>beach</i>	40.7 57.4	40.7 55.6
<i>bus</i>	38.9 45.4	58.3 52.8
<i>cafe/restaurant</i>	43.5 74.1	48.1 46.3
<i>car</i>	64.8 70.4	0.0 29.6
<i>city centre</i>	79.6 81.5	37.0 53.7
<i>forest path</i>	85.2 84.3	43.5 70.4
<i>grocery store</i>	49.1 47.2	47.2 40.7
<i>home</i>	76.9 71.3	69.4 73.1
<i>library</i>	30.6 29.6	14.8 14.8
<i>metro station</i>	93.5 92.6	1.9 24.1
<i>office</i>	73.1 91.7	57.4 53.7
<i>park</i>	32.4 37.0	28.7 15.7
<i>residential area</i>	77.8 59.3	3.7 27.8
<i>train</i>	72.2 75.0	35.2 44.4
<i>tram</i>	57.4 58.3	6.5 25.9
Overall	61.0 65.0	32.8 41.9

classification. On classifiers, SVM (cf. Section 2.4.1), GRNN (cf. Section 2.4.2) and BGRNN (cf. Section 2.4.2) were selected. Models trained independently by different feature sets were combined within a late fusion strategy, i.e., MV (cf. Section 2.6) or MSV (cf. Section 2.6). The publicly accessible database of the DCASE Challenge 2017 Task 1 [48] were used to evaluate these proposed methods. Experiments were done in clean (Section 3.3.4) and noisy (with SNR level at -10 dB, -5 dB, 0 dB, 5 dB, and 10 dB) environments (Section 3.3.5), respectively.

In the clean environment, a single model trained by COMPARE with SVM (accuracy of 61.3%) and WEF (accuracy of 63.3%) with GRNN can outperform the official baseline (accuracy of 61.0%). The fusion strategy (MV or MSV) can considerably improve the performances of models, which generates more promising results (the ones higher than the official baseline). Among these results, SVM-based late fusion (by MSV strategy) of COMPARE and WPTE reached the best accuracy of 65.0%, which significantly outperformed the official baseline ($p < .01$ by one-tailed z -test).

3. Applications

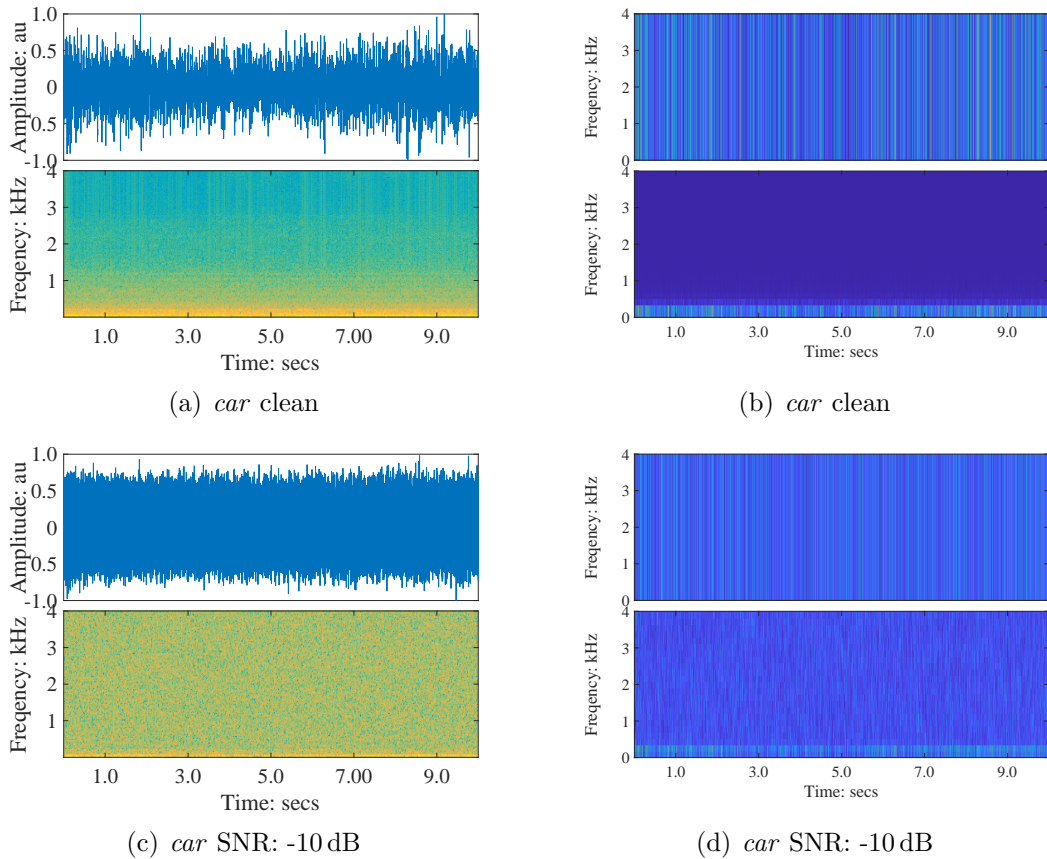


Figure 3.8: Examples of time-frequency analysis of an acoustic scene labelled as *car* in clean and noisy environment (SNR: -10 dB). Waveform (top) and spectrogram (bottom) are listed in (a) and (c). Multi-resolution time-frequency analysis by WPT (top: $J = 2$, bottom: $J = 7$) is listed in (b) and (d). Wavelet type: ‘db7’. J : decomposition level.

In the noisy environments, the contributions of wavelet features, i.e., WPTE and WEF were more obvious than those in clean environment. In particular, the fused models can still have robustness in low SNR environment (e.g., -10 dB). In such a low SNR environment, the signal is so weak that the *official baseline* cannot even work for some certain acoustic scenes (e.g., *car*, *metro station*, *residential area*, *tram*), which can be found in more detail from Table 3.24. In contrast, the best proposed system, i.e., SVM-based late fusion (by MSV strategy) of COMPARE, WPTE and WEF can significantly ($p < .001$ by one-tailed z -test) improve the class-wise accuracies for those aforementioned acoustic scenes (see Table 3.24). Figure 3.8 gives the examples of time-frequency analysis of the acoustic scene labelled as *car* by Fourier transformation and WPT in clean and noisy (SNR: -10 dB) environment, respectively. When introducing wavelet features, there is more information that

can be given to the models by using multi-resolution analysis rather than only using traditional short-time Fourier transformation (STFT). Therefore, similar to the conclusion in SnS classification (cf. 3.1), wavelet features can contribute to the classification of acoustic scenes.

It should be noted that, in the previous study [113], when fusing models learnt by different classifiers, the final performance of the system was decreased. Hence, in this study, fusion strategies, i. e., MV and MSV, were applied to features rather than classifiers. Besides, to make a direct comparison with the official baseline, there was no *data augmentation* involved in these experiments, which might limit the performances on the test set. One potential future direction is to use the state-of-the-art data augmentation techniques, e. g., *generative adversarial network* (GAN) [254], which mainly contributed to the work by the winner of DCASE Challenge 2017 Task 1 [255]. Moreover, both the baseline system and the proposed systems extracted features by averaging the left and the right channel into one channel when reading the audio recordings, which in result in losing the spacial information of the recordings. In fact, both the first [255] and second place [72] of the DCASE Challenge 2017 Task 1 used multi-channel input rather than mono to extract features. Therefore, another future work can be done at exploring the spacial information from the stereo recordings.

Conclusion

As an increasingly developing field in *Artificial Intelligence* (AI), *computer audition* (CA) can facilitate intelligent systems to hear like humans or beyond the human hearing frontier, i. e., 20–20 000 Hz [256]. Automatic general audio signal classification (AGASC), aims to leverage methods in *signal processing* and *machine learning* to build an intelligent system hearing more general audio signals (e. g., snore sound, bird sound, acoustic scene) rather than speech or music. Compared to the study on speech or music, AGASC is a young research area. The challenges of three typical tasks in AGASC, i. e., *snore sound classification*, *bird sound classification*, and *acoustic scene classification* were proposed (cf. Chapter 1), and addressed (cf. Chapter 3) by the methods described (cf. Chapter 2) in this thesis.

In this chapter, a summary will be given in Section 4.1. Section 4.2 will sketch some future directions based on the limitations of the current research work.

4.1 Summary

After an introduction of the general research background, challenges, and contributions in Chapter 1, the proposed methodologies used in this thesis were described in Chapter 2 (theoretical aspects) and evaluated in Chapter 3 (empirical aspects). The work achieved in each of the aforementioned three tasks can be summarised as follows:

(1) A comprehensive comparison on features and classifiers for SnS classification was given. In Section 2.1, firstly, the previously used ones in the snore sound area (e. g., Formants, SFFs, SERs, MFCCs) were introduced. Then, an integration of human expert designed temporal and spectral features (COMPARE) extracted by the OPENSMILE toolkit was described. Finally, *wavelet features*, i. e., WTE, WPTE, and WEF were proposed. The classifiers (Section 2.4), including the classical models (e. g., NB, k -NN, SVM, RF, MLP) and the state-of-the-art (ELM, KELM, SAE) were selected. The experimental results were given in Section 3.1.4. The *wavelet features*

were found to be efficient in SnS classification task. A late fusion of SVM and ELM models trained individually by wavelet features can reach an excellent performance. Moreover, a combination of wavelet features and the BoAW approach (Section 2.2.2) was found to considerably improve the capacity of the classifier (NB) to recognise SnS, when comparing the features extracted as *functionals* (Section 2.2.1). The work was evaluated in Section 3.1.5. More precisely, the best proposed method for SnS classification, i. e., a WEF feature set enhanced via BoAW (trained by NB classifier) beat the winner of the INTERSPEECH COMPARE challenge 2017 Snoring sub-challenge. Encouragingly, it was found that two proposed methods in this thesis can even outperform some other participants' methods using *deep learning* [163,164,257], and *evolutional feature selection* [164] (Section 3.1.5).

(2) To reduce the human annotation work from a large amount of unlabelled bird sound data, two popular algorithms of AL, i. e., SI-AL and LCS-AL were proposed (Section 2.5.2). Compared with PL (Section 2.5.1), AL (except SI-AL by an SVM classifier) algorithms can be efficient in finding 'most informative' data from the unlabelled pool data set, which in result can considerably reduce the need of human annotation when improving the trained model's recognition performance to a certain level. Particularly, LCS-AL was found to be superior to SI-AL in reducing the human annotation work. In addition, LCS-AL was demonstrated to be more stable and robust than SI-AL when run in replicated experiments (Section 3.2.5). It was also found that, when selecting a suitable classifier, e. g., KELM, the performance and robustness of AL algorithms can be dramatically enhanced (Section 3.2.5). These successful experimental results will stimulate the work in data enrichment for animal sound (e. g., bird sound) recognition, which in nature has to handle much more data unlabelled than labelled by human experts.

(3) Motivated by the success of wavelet features in SnS classification (Section 3.1), WPTE (Section 2.1.6) and WEF (Section 2.1.6) were introduced to the ASC task (Section 3.3). As to features, the COMPARE feature set (Section 2.1.5) was combined with the two aforementioned wavelet features by a late fusion strategy, i. e., MV (Section 2.6) or MSV (Section 2.6). As to classifiers, static modelling (SVM, cf. Section 2.4.1) and sequential modelling (GRNN, cf. Section 2.4.2 and BGRNN, cf. Section 2.4.2) were used. The experimental results showed the effectiveness and robustness of the proposed systems in both clean (Section 3.3.4) and noisy environments (Section 3.3.5). Particularly, singly using wavelet features can be comparable to the COMPARE feature set (which has much larger size of feature dimension and are mainly based on Fourier transformation). Furthermore, when combining wavelet features with the COMPARE feature set, most of the proposed systems can be superior to the official baseline system in the DCASE Challenge 2017 Task 1 [48], particularly in noisy environments (Section 3.3.5).

4.2 Outlook

The proposed methods had been successfully evaluated by publicly accessible databases in this thesis, which advanced the state-of-the-art in the area of AGASC. However, the accomplished work reaches only the tip of the iceberg of AGASC. There are several future directions along which the line of research can be investigated in the future.

First, there are not so many *deep learning* architectures involved in this thesis; the main reason is due to the limited size of the databases as are used currently. It is reasonable to think that deep architectures can learn richer information from the inputs than shallow architectures when the data size becomes big enough. Therefore, the proposed methods will be optimised to be implemented in deep models with databases having large size, e. g., Bird Task by LifeCLEF [241], AudioSet by Google [258].

Then, all the acoustic feature proposed in this thesis are hand crafted, which needs human expert knowledge. Usually, designing suitable features for a specific domain is expensive and time-consuming. Recently, some emerging techniques using *convolutional neural networks* (CNNs) to learn feature representations from the spectrogram/scalogram of audio signal have been popularly investigated in classification of snore sounds [163, 164], bird sounds [39, 40], and acoustic scenes [70, 165]. One direction of future work can be leveraging *transfer learning* [259], and *multi-task learning* [260] to make the system automatically learn deep robust representations from images generated from the general audio signals.

Moreover, AL still needs some human expert annotation work in its paradigm. The state-of-the-art semi-supervised learning [261], cooperative learning [262] should be investigated and developed to help improve the efficiency of the whole data enrichment process for AGASC tasks. In addition, data augmentation, specifically for deep learning, is an important phase when the data size is limited. Another direction in this line is to use *generative adversarial networks* (GANs) [254] to play the role in data in the augmentation phase.

Overall, hopefully more attention could be attracted to the field presented by this thesis and the relevant work could stimulate future research.

A

Bird Species

The original MNB bird sound database contains 6 487 audio recordings from 273 species (subspecies) of birds. However, there are many species only including several few audio recordings. To make an efficient study on training models and evaluating algorithms, the species which have less than 20 audio recordings were eliminated. The name and number of instances (audio recordings) for each bird species selected are listed in Table A.1. The selected subset of the MNB bird sound database includes 86 species of birds and 5 060 audio recordings.

Table A.1: *Latin* name and number of instances for each bird species.

<i>Latin</i> Name	# Instances	<i>Latin</i> Name	# Instances	<i>Latin</i> Name	# Instances
<i>Acrocephalus arundinaceus</i>	59	<i>Acrocephalus palustris</i>	37	<i>Acrocephalus scirpaceus</i>	59
<i>Aegolius funereus</i>	50	<i>Alauda arvensis</i>	24	<i>Anthus petrosus</i>	24
<i>Anthus pratensis</i>	38	<i>Anthus trivialis</i>	44	<i>Asio otus</i>	70
<i>Athene noctua</i>	22	<i>Botaurus stellaris</i>	22	<i>Caprimulgus europaeus</i>	20
<i>Carpodacus erythrinus</i>	31	<i>Certhia brachydactyla</i>	42	<i>Certhia familiaris</i>	22
<i>Chloris chloris</i>	34	<i>Chroicocephalus ridibundus</i>	21	<i>Corvus corax</i>	21
<i>Cyanistes caeruleus</i>	36	<i>Dendrocoptes major</i>	53	<i>Dendrocoptes medius</i>	50
<i>Dendrocoptes minor</i>	28	<i>Dryocopus martius</i>	53	<i>Emberiza calandra</i>	68
<i>Emberiza citrinella</i>	86	<i>Emberiza hortulana</i>	279	<i>Emberiza rustica</i>	32
<i>Emberiza schoeniclus</i>	119	<i>Erethacus rubecula</i>	26	<i>Ficedula hypoleuca</i>	44
<i>Ficedula parva</i>	35	<i>Fringilla coelebs</i>	261	<i>Fringilla montifringilla</i>	23
<i>Fulica atra</i>	59	<i>Gallinula chloropus</i>	28	<i>Garrulus glandarius</i>	24
<i>Hirundo rustica</i>	23	<i>Jynx torquilla</i>	20	<i>Locustella naevia</i>	23
<i>Lophophanes cristatus</i>	64	<i>Lullula arborea</i>	22	<i>Luscinia luscinia</i>	133
<i>Luscinia megarhynchos</i>	179	<i>Motacilla alba</i>	24	<i>Muscicapa striata</i>	23
<i>Oriolus oriolus</i>	23	<i>Parus major</i>	145	<i>Passer domesticus</i>	23
<i>Passer montanus</i>	22	<i>Periparus ater</i>	187	<i>Phalacrocorax carbo</i>	29
<i>Phoenicurus ochruros</i>	43	<i>Phoenicurus phoenicurus</i>	155	<i>Phylloscopus bonelli</i>	64
<i>Phylloscopus canariensis</i>	38	<i>Phylloscopus collybita</i>	132	<i>Phylloscopus ibericus</i>	129
<i>Phylloscopus sibilatrix</i>	34	<i>Phylloscopus trochilus</i>	133	<i>Pica pica</i>	20
<i>Picus canus</i>	54	<i>Picus viridis</i>	28	<i>Podiceps cristatus</i>	27
<i>Podiceps grisegena</i>	31	<i>Poecetes montanus</i>	20	<i>Porzana parva</i>	23
<i>Porzana porzana</i>	66	<i>Prunella modularis</i>	26	<i>Rallus aquaticus</i>	113
<i>Regulus ignicapilla</i>	20	<i>Regulus regulus</i>	20	<i>Sarcicola rubetra</i>	79
<i>Sitta europaea</i>	37	<i>Strix aluco</i>	34	<i>Sylvia atricapilla</i>	110
<i>Sylvia borin</i>	54	<i>Sylvia communis</i>	76	<i>Sylvia curruca</i>	43
<i>Sylvia melanocephala</i>	50	<i>Sylvia nisoria</i>	39	<i>Troglodytes troglodytes</i>	58
<i>Turdus merula</i>	156	<i>Turdus philomelos</i>	124	<i>Turdus pilaris</i>	54
<i>Turdus viscivorus</i>	61	<i>Tyto alba</i>	25		

List of Acronyms

AAD	Audio Activity Detection
AE	Autoencoder
AGASC	Automatic General Audio Signal Classification
AI	Artificial Intelligence
AL	Active Learning
AR	Autoregressive
ASC	Acoustic Scene Classification
ASR	Automatic Speech Recognition
BGRNN	Bidirectional Gated Recurrent Neural Network
BP	Backpropagation
BPTT	Backpropagation Through Time
BRNN	Bidirectional Recurrent Neural Network
BoAW	Bag-of-Audio-Words
CA	Computer Audition
CASA	Computational Auditory Scene Analysis
CNN	Convolutional Neural Network
COMPARE	Computational Paralinguistics Challenge
DAE	Denoising Autoencoder
DCASE	Detection and Classification of Acoustic Scenes and Events
DISE	Drug Induced Sleep Endoscopy
DL	Deep Learning

List of Acronyms

DNN	Deep Neural Network
DTW	Dynamic Time Warping
DWT	Discrete Wavelet Transformation
ELM	Extreme Learning Machine
ENT	Ear, Nose and Throat
F0	Fundamental Frequency
FFT	Fast Fourier Transformation
FNN	Feed-Forward Neural Network
GAN	Generative Adversarial Network
GMM	Gaussian Mixture Model
GRNN	Gated Recurrent Neural Network
GRU	Gated Recurrent Unit
HMM	Hidden Markov Model
HNR	Harmonics to Noise Ratio
KELM	Kernel-based Extreme Learning Machine
LCS-AL	Least-Condence-Score-based AL
LLD	Low-Level Descriptor
LPC	Linear Predictive Coding
LPCC	Linear Prediction Cepstral Coefficient
LSTM	Long Short-Term Memory
MAP	Maximum A Posteriori
MFCC	Mel-frequency Cepstral Coefficient
MIML	Multi-Instance Multi-Label
MIR	Music Information Retrieval
MLP	Multilayer Perceptron
MNB	Museum für Naturkunde Berlin
MPSSC	Munich Passau Snore Sound Corpus
MSE	Mean Squared Error
MSV	Margin Sampling Voting
MV	Majority Voting

NB	Naïve Bayes
NMF	Nonnegative Matrix Factorization
OSA	Obstructive Sleep Apnea
PL	Passive Learning
PSG	Polysomnography
RASTA	Representations Relative Spectra
RASTA-PLP	Representations Relative Spectra Perceptual Linear Prediction
RBF	Radial Basis Function
RF	Random Forest
RMSE	Root Mean Square Energy
RNN	Recurrent Neural Network
RoP	Roll-off Point
SAE	Stacked Autoencoder
SCG	Scaled Conjugate Gradient
SER	Subband Energy Ratio
SFF	Spectral Frequency Feature
SGD	Stochastic Gradient Descent
SHS	Subharmonic Summation
SI-AL	Sparse Instance-based AL
SLFNN	Single Hidden Layer Feed-Forward Neural Network
SMO	Sequential Minimal Optimisation
SOM	Self-Organising Map
SSL	Semi-Supervised Learning
STFT	Short-Time Fourier Transformation
SVM	Support Vector Machine
SnS	Snore Sound
UAR	Unweighted Average Recall
VQ	Vector Quantisation
WAR	Weighted Average Recall
WEF	Wavelet Energy Feature

List of Acronyms

WKELM.....	Weighted Kernel-based Extreme Learning Machine
WKPLS.....	Weighted Kernel Partial Least Squares
WPT.....	Wavelet Packet Transformation
WPTE.....	Wavelet Packet Transform Energy
WT.....	Wavelet Transformation
WTE.....	Wavelet Transform Energy
ZCR.....	Zero-Crossing Rate
<i>k</i> -NN.....	<i>k</i> -Nearest-Neighbour

List of Symbols

Acoustical Low-Level Descriptors

i	index of LPC coefficient, root in AR model, formant frequency
p	linear predictor order
a_i	i -the LPC coefficient
r_i	i -th root in AR model
z	point in z -domain complex plane
F_s	sampling frequency of audio signal
F_i	i -th formant frequency
F_{peak}	peak frequency
F_{centre}	centre frequency
F_{mean}	mean frequency
n	index of FFT point
N	number of FFT points
f_n	frequency at n -th point FFT
X_{f_n}	absolute value of the single-sided amplitude by FFT at frequency of f_n
F_h	highest frequency of the spectrum of audio signal
m	index of subband in the spectrum
$F_{mean-sub(m)}$	mean frequency in m -th subband spectrum
F_b	frequency to set subband of the whole spectrum
$SER_{(m)}$	energy ratio at m -th subband

f	real scale frequency
$f_{(Mel)}$	Mel-scale frequency
n'	index of value in WT or WPT functions
j	scale or index of decomposition level of WT or WPT
k	index of subband of WT or WPT
$\psi(\cdot)$	wavelet function
$\phi(\cdot)$	scaling function
\mathbf{V}	signal space
\mathbf{E}_j	vector of energy percentage at j -th decomposition level by WT
\mathbf{w}_j	vector of coefficients generated by WT at j -th decomposition level
J_{max}	maximum level for wavelet decomposition by WT or WPT
n''	index of the coefficients generated by WPT
$\mathbf{w}_{j,k,n''}$	vector of coefficients calculated by WPT from the analysed signal at the subspace $\mathbf{V}_{j,k}$
$\hat{E}_{j,k}$	normalised filter bank energy of k -th subband at j -th decomposition level by WPT
$N_{j,k}$	total number of coefficients in k -th subband at j -th decomposition level by WPT

Functionals and Bag-of-Audio-Words

n	index of points in time series
y	time series
$x(n)$	time series
N	number of points in time series
x_{max}	maximum value
x_{min}	minimum value
x_{mean}	arithmetic mean
a	linear regression slope
b	linear regression offset
\hat{y}	approximated line

\hat{e}^2	quadratic error between \hat{y} and y
k	number of initial centres for k -mean clustering
i	index of centre or cluster
j	index of centre or cluster
c_i	i -th centre
\mathcal{S}_i	i -th cluster
x	data point
\mathcal{X}	data set
$D(x)$	shortest distance from x to the closest centre that has already been chosen

Feature Normalisation

\mathbf{x}	feature vector
\mathbf{x}'	normalised feature vector
n	index of value in \mathbf{x}
N	number of values in \mathbf{x}
a	linear scaling value
b	linear scaling value
$\mu_{\mathbf{x}}$	arithmetic mean calculated from \mathbf{x}
x_{max}	maximum value in \mathbf{x}
x_{min}	minimum value in \mathbf{x}
$\sigma_{\mathbf{x}}$	standard deviation calculated from \mathbf{x}

Classification

\mathbf{x}	feature vector
y	label or prediction
\mathbb{R}^d	d -dimensional feature space
\mathcal{Y}	label variable
κ	number of classes

Classical Models

- x value in \mathbf{x}
- d index of value in \mathbf{x}
- $P(y|\mathbf{x})$ conditional probability
- $P(y)$ relative frequency
- k number of nearest neighbours in k -NN
- \mathbf{x}_t a given instance in test set
- \mathbf{x}_σ nearest neighbour of \mathbf{x}_t
- y_σ label of \mathbf{x}_σ
- i index of instance in training set
- j index of instance in training set
- n number of instances in training set
- α Lagrange multiplier
- C_s pre-defined parameter for SVM
- $K(\cdot)$ kernel function
- γ pre-defined parameter for kernel
- c pre-defined parameter for kernel
- \hat{d} pre-defined parameter for kernel
- $f(\cdot)$ decision function of SVM
- b bias of SVM decision function

Deep Learning Models

- l index of layer
- \mathbf{h}^l output of l -th layer
- \mathbb{R}^n n -dimensional feature space
- \mathbb{R}^m m -dimensional feature space
- \mathbf{W} weight matrix
- \mathbf{b} bias vector

$\mathcal{F}(\cdot)$	activation function
\mathbf{y}	actual output
\mathbf{t}	target output
i	index of output node
j	index of output node
$\mathcal{L}(\cdot)$	loss function
$\boldsymbol{\theta}$	parameters of the network
τ	iteration step
η	learning rate
μ	momentum term
\mathbf{a}	new representation of \mathbf{x} by encoder
\mathbf{x}'	reconstruction of \mathbf{x} by decoder
\mathcal{F}'	activation function in decoder stage
\mathbf{W}'	weight matrix in decoder stage
\mathbf{b}'	bias vector in decoder stage
j	index of input vector
N	number of input vectors
L_2	L_2 regularisation term
$SP(\cdot)$	sparsity regularisation term
ρ	sparsity level
n	number of hidden nodes
$h_k(\cdot)$	activation value of k -th node
$\hat{\rho}_k$	average activation value of k -th node
α	parameter for L_2
β	parameter for $SP(\cdot)$
t	time step
\mathbf{U}	recurrent weight matrix
\mathbf{z}	update gate in GRU
\mathbf{r}	reset gate in GRU
$\tilde{\mathbf{h}}$	candidate activation in GRU

\vec{h} forward hidden layer activation
 \overleftarrow{h} backward hidden layer activation
 \vec{U} forward recurrent weight matrix
 \overleftarrow{U} backward recurrent weight matrix
 \vec{b} forward bias vector
 \overleftarrow{b} backward bias vector

Extreme Learning Models

D dimension of input feature vector
 L number of hidden nodes
 l index of hidden node
 \mathbf{a} input weight vector
 b bias
 \mathbf{w} output weight vector
 $f(\cdot)$ output function at one node
 $\mathbf{h}(\cdot)$ output of hidden layer
 $\mathcal{G}(\cdot)$ activation function
 N number of training examples
 \mathbf{X} training examples
 \mathbf{T} target matrix
 \mathbf{H} output matrix of hidden layer
 \mathbf{W} output weight matrix
 \mathbf{H}^\dagger Moore-Penrose generalised inverse of \mathbf{H}
 \mathbf{I} identity matrix
 C_e pre-defined parameter for ELM
 \mathbf{f} output vector
 m index of output node
 M number of output nodes
 $\mathbf{\Omega}_{ELM}$ kernel matrix for ELM

Data Enrichment

\mathcal{L}	labelled data
\mathcal{U}	unlabelled data
\mathcal{C}	trained model
K	number of selected samples for human annotation in each iteration of PL or AL
\mathcal{D}_K	selected subset with K samples for human annotation in each iteration of PL or AL
\mathbf{x}	instance
$Q_{SI}(\cdot)$	query function by SI-AL
$\hat{y}_{\mathbf{x}}$	predicted label of instance \mathbf{x}
\mathcal{Y}_{sparse}	a set of sparse classes
i	index of sparse class
\mathcal{Y}_i	label of i -th sparse class
$N_{\mathcal{Y}_i}$	number of instances that belong to \mathcal{Y}_i class
λ_s	sparse factor in SI-AL
N_s	total number of sparse classes
N_{max}	number of instances that belong to one certain class which occupies the biggest proportion in the whole data
$Q_{LCS}(\cdot)$	query function by LCS-AL
$CS(\cdot)$	function to calculate confidence score
$P_{\mathcal{C}}(\cdot)$	posterior probability under the trained model \mathcal{C}
\hat{y}_1	predicted label corresponding to the first highest posterior probability
\hat{y}_2	predicted label corresponding to the second highest posterior probability
λ_c	pre-defined factor in LCS-AL
$N_{\mathcal{U}}$	number of instances in \mathcal{U}

Late Fusion

i	index of model
j	index of class
\tilde{y}	final prediction made by MV or MSV strategy
$d_{i,j}$	decision value of i -th model for j -th class
\mathcal{N}	number of models
N_c	number of classes
\mathcal{Y}_i	prediction of i -th model
\mathcal{Y}_j	label of j -th class
$\mathcal{W}_{i,\mathcal{Y}_i}$	calibration weight for i -th model to make its prediction as \mathcal{Y}_i
\mathcal{M}_i	margin sampling value of i -th model

Evaluation Metrics

i	index of class
\tilde{N}_i	number of correctly predicted instances for i -th class
N_i	total number of instances labelled as i -th class
λ_i	weight for i -th class
N_c	number of classes
N	total number of instances
z	standard score for z -test
m_A	measure value of system A
m_B	measure value of system B
m	arithmetic mean value of m_A and m_B
$\Phi(\cdot)$	standard normal cumulative distribution function
t	test static for Student's t -test
N_A	sample size of measures in system A
N_B	sample size of measures in system B
\bar{m}_A	sample mean of measures in system A
\bar{m}_B	sample mean of measures in system B

- σ_A sample standard deviation of measures in system A
 σ_B sample standard deviation of measures in system B
 p p -value calculated by z -test or Student's t -test
 α significance level

Bibliography

- [1] D. S. Pallett, “A look at NIST’s benchmark ASR tests: Past, present, and future,” in *Proc. ASRU*, St Thomas, VI, USA, 2003, pp. 483–488.
- [2] J. Futrelle and J. S. Downie, “Interdisciplinary research issues in music information retrieval: ISMIR 2000–2002,” *Journal of New Music Research*, vol. 32, no. 2, pp. 121–131, 2003.
- [3] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, “Detection and classification of acoustic scenes and events,” *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.
- [4] D. Pevernagie, R. M. Aarts, and M. De Meyer, “The acoustics of snoring,” *Sleep Medicine Reviews*, vol. 14, no. 2, pp. 131–144, 2010.
- [5] P. J. Strollo Jr and R. M. Rogers, “Obstructive sleep apnea,” *New England Journal of Medicine*, vol. 334, no. 2, pp. 99–104, 1996.
- [6] M. Koskenvuo, J. Kaprio, T. Telakivi, M. Partinen, K. Heikkilä, and S. Sarna, “Snoring as a risk factor for ischaemic heart disease and stroke in men,” *British Medical Journal*, vol. 294, no. 6563, pp. 16–19, 1987.
- [7] W. W. Schmidt-Nowara, D. B. Coultas, C. Wiggins, B. E. Skipper, and J. M. Samet, “Snoring in a hispanic-american population: risk factors and association with hypertension and other morbidity,” *Archives of Internal Medicine*, vol. 150, no. 3, pp. 597–601, 1990.
- [8] R. D’alessandro, C. Magelli, G. Gamberini, S. Bacchelli, E. Cristina, B. Magnani, and E. Lugaresi, “Snoring every night as a risk factor for myocardial infarction: a case-control study,” *British Medical Journal*, vol. 300, no. 6739, pp. 1557–1558, 1990.

- [9] T. Seppälä, M. Partinen, A. Penttilä, R. Aspholm, E. Tiainen, and A. Kaukianen, “Sudden death and sleeping history among finnish men,” *Journal of Internal Medicine*, vol. 229, no. 1, pp. 23–28, 1991.
- [10] P. E. Peppard, T. Young, J. H. Barnet, M. Palta, E. W. Hagen, and K. M. Hla, “Increased prevalence of sleep-disordered breathing in adults,” *American Journal of Epidemiology*, vol. 177, no. 9, pp. 1006–1014, 2013.
- [11] K. K. Li, “Surgical therapy for adult obstructive sleep apnea,” *Sleep Medicine Reviews*, vol. 9, no. 3, pp. 201–209, 2005.
- [12] H.-C. Lin, M. Friedman, H.-W. Chang, and B. Gurpinar, “The efficacy of multilevel surgery of the upper airway in adults with obstructive sleep apnea/hypopnea syndrome,” *The Laryngoscope*, vol. 118, no. 5, pp. 902–908, 2008.
- [13] A. V. Vroegop, O. M. Vanderveken, A. N. Boudewyns, J. Scholman, V. Saldien, K. Wouters, M. J. Braem, P. H. Van de Heyning, and E. Hamans, “Drug-induced sleep endoscopy in sleep-disordered breathing: Report on 1,249 cases,” *The Laryngoscope*, vol. 124, no. 3, pp. 797–802, 2014.
- [14] C. Janott, B. Schuller, and C. Heiser, “Acoustic information in snoring noise,” *HNO*, vol. 65, no. 2, pp. 107–116, 2017.
- [15] K. Qian, Y. Fang, Z. Xu, and H. Xu, “Comparison of two acoustic features for classification of different snore signals,” *Chinese Journal of Electron Devices*, vol. 36, no. 4, pp. 455–459, 2013.
- [16] K. Qian, Z. Xu, H. Xu, and B. P. Ng, “Automatic detection of inspiration related snoring signals from original audio recording,” in *Proc. ChinaSIP*, Xi’an, China, 2014, pp. 95–99.
- [17] K. Qian, Z. Xu, H. Xu, Y. Wu, and Z. Zhao, “Automatic detection, segmentation and classification of snore related signals from overnight audio recording,” *IET Signal Processing*, vol. 9, no. 1, pp. 21–29, 2015.
- [18] K. Qian, C. Janott, Z. Zhang, C. Heiser, and B. Schuller, “Wavelet features for classification of snore sounds,” in *Proc. ICASSP*, Shanghai, China, 2016, pp. 221–225.
- [19] M. Schmitt, C. Janott, V. Pandit, K. Qian, C. Heiser, W. Hemmert, and B. Schuller, “A bag-of-audio-words approach for snore sounds excitation localisation,” in *Proc. ITG Speech Communication*, Paderborn, Germany, 2016, pp. 230–234.

-
- [20] K. Qian, C. Janott, J. Deng, C. Heiser, W. Hohenhorst, M. Herzog, N. Cummins, and B. Schuller, “Snore sound recognition: on wavelets and classifiers from deep nets to kernels,” in *Proc. EMBC*, Jeju Island, Korea, 2017, pp. 3737–3740.
- [21] K. Qian, C. Janott, V. Pandit, Z. Zhang, C. Heiser, W. Hohenhorst, M. Herzog, W. Hemmert, and B. Schuller, “Classification of the excitation location of snore sounds in the upper airway by acoustic multi-feature analysis,” *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 8, pp. 1731–1741, 2017.
- [22] A. Balmford, R. E. Green, and M. Jenkins, “Measuring the changing state of nature,” *Trends in Ecology & Evolution*, vol. 18, no. 7, pp. 326–330, 2003.
- [23] C. Parmesan and G. Yohe, “A globally coherent fingerprint of climate change impacts across natural systems,” *Nature*, vol. 421, no. 6918, pp. 37–42, 2003.
- [24] Ç. H. Şekercioglu, G. C. Daily, and P. R. Ehrlich, “Ecosystem consequences of bird declines,” *Proceedings of the National Academy of Sciences*, vol. 101, no. 52, pp. 18 042–18 047, 2004.
- [25] A. Gasc, J. Sueur, F. Jiguet, V. Devictor, P. Grandcolas, C. Burrow, M. Depraetere, and S. Pavoine, “Assessing biodiversity with sound: Do acoustic diversity indices reflect phylogenetic and functional diversities of bird communities?” *Ecological Indicators*, vol. 25, pp. 279–287, 2013.
- [26] A. L. McIlraith and H. C. Card, “Birdsong recognition using backpropagation and multivariate statistics,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2740–2748, 1997.
- [27] P. Somervuo, A. Harma, and S. Fagerlund, “Parametric representations of bird sounds for automatic species recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 6, pp. 2252–2263, 2006.
- [28] Z. Chen and R. C. Maher, “Semi-automatic classification of bird vocalizations using spectral peak tracks,” *The Journal of the Acoustical Society of America*, vol. 120, no. 5, pp. 2974–2984, 2006.
- [29] S. Fagerlund, “Bird species recognition using support vector machines,” *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 64–64, 2007.
- [30] A. Selin, J. Turunen, and J. T. Tantt, “Wavelets in recognition of bird sounds,” *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 141–141, 2007.

- [31] C.-H. Lee, C.-C. Han, and C.-C. Chuang, “Automatic classification of bird species from their sounds using two-dimensional cepstral coefficients,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1541–1550, 2008.
- [32] C.-H. Lee, S.-B. Hsu, J.-L. Shih, and C.-H. Chou, “Continuous birdsong recognition using gaussian mixture modeling of image shape features,” *IEEE Transactions on Multimedia*, vol. 15, no. 2, pp. 454–464, 2013.
- [33] J. R. Heller and J. D. Pinezich, “Automatic recognition of harmonic bird sounds using a frequency track extraction algorithm,” *The Journal of the Acoustical Society of America*, vol. 124, no. 3, pp. 1830–1837, 2008.
- [34] P. Jančovič and M. Köküer, “Automatic detection and recognition of tonal bird sounds in noisy environments,” *EURASIP Journal on Advances in Signal Processing*, vol. 2011, no. 1, pp. 1–10, 2011.
- [35] F. Briggs, B. Lakshminarayanan, L. Neal, X. Z. Fern, R. Raich, S. J. Hadley, A. S. Hadley, and M. G. Betts, “Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach,” *The Journal of the Acoustical Society of America*, vol. 131, no. 6, pp. 4640–4650, 2012.
- [36] P. Jančovič and M. Köküer, “Acoustic recognition of multiple bird species based on penalized maximum likelihood,” *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1585–1589, 2015.
- [37] M. Lasseck, “Large-scale identification of birds in audio recordings.” in *CLEF Working Notes*, Sheffield, UK, 2014, pp. 643–653.
- [38] D. Stowell and M. D. Plumbley, “Audio-only bird classification using unsupervised feature learning,” in *CLEF Working Notes*, Sheffield, UK, 2014, pp. 673–684.
- [39] B. P. Tóth and B. Czéba, “Convolutional neural networks for large-scale bird song classification in noisy environment,” in *CLEF Working Notes*, Evora, Portugal, 2016, pp. 560–568.
- [40] E. Sprengel, M. Jaggi, Y. Kilcher, and T. Hofmann, “Audio based bird species identification using deep learning techniques,” in *CLEF Working Notes*, Evora, Portugal, 2016, pp. 547–559.
- [41] K. J. Piczak, “Recognizing bird species in audio recordings using deep convolutional neural networks,” in *CLEF Working Notes*, Evora, Portugal, 2016, pp. 534–543.

-
- [42] J. A. Kogan and D. Margoliash, “Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden markov models: A comparative study,” *The Journal of the Acoustical Society of America*, vol. 103, no. 4, pp. 2185–2196, 1998.
- [43] L. Ranjard and H. A. Ross, “Unsupervised bird song syllable classification using evolving neural networks,” *The Journal of the Acoustical Society of America*, vol. 123, no. 6, pp. 4358–4368, 2008.
- [44] R. O. Tachibana, N. Oosugi, and K. Okanoya, “Semi-automatic classification of birdsong elements using a linear support vector machine,” *PloS One*, vol. 9, no. 3, pp. 1–8, 2014.
- [45] L. N. Tan, A. Alwan, G. Kossan, M. L. Cody, and C. E. Taylor, “Dynamic time warping and sparse representation classification for birdsong phrase classification using limited training data,” *The Journal of the Acoustical Society of America*, vol. 137, no. 3, pp. 1069–1080, 2015.
- [46] D. Braha, *Data Mining for Design and Manufacturing: Methods and Applications*. Dordrecht, Netherlands: Springer Science+Business Media B.V., 2001.
- [47] D. Wang and G. J. Brown, *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Piscataway, NJ, US: Wiley-IEEE Press, 2006.
- [48] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, “DCASE2017 challenge setup: Tasks, datasets and baseline system,” in *Proc. DCASE Workshop*, Munich, Germany, 2017, pp. 85–92.
- [49] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley, “Detection and classification of acoustic scenes and events: Outcome of the dcase 2016 challenge,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 2, pp. 379–393, 2018.
- [50] M. Bugalho, J. Portelo, I. Trancoso, T. Pellegrini, and A. Abad, “Detecting audio events for semantic video search,” in *Proc. INTERSPEECH*, Brighton, UK, 2009.
- [51] A. J. Eronen, V. T. Peltonen, J. T. Tuomi, A. P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, “Audio-based context recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 321–329, 2006.

- [52] D. Stowell and D. Clayton, “Acoustic event detection for multiple overlapping similar sources,” in *Proc. WASPAA*, New Paltz, NY, US, 2015, pp. 1–5.
- [53] S. Goetze, J. Schroder, S. Gerlach, D. Hollosi, J.-E. Appell, and F. Wallhoff, “Acoustic monitoring and localization for social care,” *Journal of Computing Science and Engineering*, vol. 6, no. 1, pp. 40–50, 2012.
- [54] R. Radhakrishnan, A. Divakaran, and A. Smaragdis, “Audio analysis for surveillance applications,” in *Proc. WASPAA*, New Paltz, NY, USA, 2005, pp. 158–161.
- [55] A. Rabaoui, M. Davy, S. Rossignol, and N. Ellouze, “Using one-class svms and wavelets for audio surveillance,” *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 4, pp. 763–775, 2008.
- [56] X. Wu, H. Gong, P. Chen, Z. Zhong, and Y. Xu, “Surveillance robot utilizing video and audio information,” *Journal of Intelligent and Robotic Systems*, vol. 55, no. 4-5, pp. 403–421, 2009.
- [57] 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.
- [58] J.-J. Aucouturier, B. Defreville, and F. Pachet, “The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music,” *The Journal of the Acoustical Society of America*, vol. 122, no. 2, pp. 881–891, 2007.
- [59] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, “Audio context recognition using audio event histograms,” in *Proc. EUSIPCO*, Aalborg, Denmark, 2010, pp. 1272–1276.
- [60] 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.
- [61] V. Bisot, R. Serizel, S. Essid, and G. Richard, “Nonnegative feature learning methods for acoustic scene classification,” in *Proc. DCASE Workshop*, Munich, Germany, 2017, pp. 22–26.
- [62] B. Elizalde, A. Kumar, A. Shah, R. Badlani, E. Vincent, B. Raj, and I. Lane, “Experiments on the DCASE Challenge 2016: Acoustic scene classification and sound event detection in real life recording,” in *Proc. DCASE Workshop*, Budapest, Hungary, 2016, pp. 20–24.

-
- [63] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [64] Q. Kong, I. Sobieraj, W. Wang, and M. Plumbley, “Deep neural network baseline for DCASE Challenge 2016,” in *Proc. DCASE Workshop*, Budapest, Hungary, 2016, pp. 50–54.
- [65] E. Marchi, D. Tonelli, X. Xu, F. Ringeval, J. Deng, S. Squartini, and B. Schuller, “Pairwise decomposition with deep neural networks and multiscale kernel subspace learning for acoustic scene classification,” in *Proc. DCASE Workshop*, Budapest, Hungary, 2016, pp. 65–69.
- [66] Y. Xu, Q. Huang, W. Wang, and M. D. Plumbley, “Hierarchical learning for DNN-based acoustic scene classification,” in *Proc. DCASE Workshop*, Budapest, Hungary, 2016, pp. 105–109.
- [67] M. Valenti, A. Diment, G. Parascandolo, S. Squartini, and T. Virtanen, “DCASE 2016 acoustic scene classification using convolutional neural networks,” in *Proc. DCASE Workshop*, Budapest, Hungary, 2016, pp. 95–99.
- [68] S. H. Bae, I. Choi, and N. S. Kim, “Acoustic scene classification using parallel combination of LSTM and CNN,” in *Proc. DCASE Workshop*, Budapest, Hungary, 2016, pp. 11–15.
- [69] J. Abeßer, S. I. Mimitakis, R. Gräfe, and H. Lukashevich, “Acoustic scene classification by combining autoencoder-based dimensionality reduction and convolutional neural networks,” in *Proc. DCASE Workshop*, Munich, Germany, 2017, pp. 7–11.
- [70] S. Amiriparian, M. Freitag, N. Cummins, and B. Schuller, “Sequence to sequence autoencoders for unsupervised representation learning from audio,” in *Proc. DCASE Workshop*, Munich, Germany, 2017, pp. 17–21.
- [71] E. Fonseca, R. Gong, D. Bogdanov, O. Slizovskaia, E. Gomez, and X. Serra, “Acoustic scene classification by ensembling gradient boosting machine and convolutional neural networks,” in *Proc. DCASE Workshop*, Munich, Germany, 2017, pp. 37–41.
- [72] Y. Han and J. Park, “Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification,” in *Proc. DCASE Workshop*, Munich, Germany, 2017, pp. 46–50.
- [73] J.-W. Jung, H.-S. Heo, I.-H. Yang, S.-H. Yoon, H.-J. Shim, and H.-J. Yu, “DNN-based audio scene classification for DCASE2017: Dual input features, balancing cost, and stochastic data duplication,” in *Proc. DCASE Workshop*, Munich, Germany, 2017, pp. 59–63.

- [74] S. Park, S. Mun, Y. Lee, and H. Ko, “Acoustic scene classification based on convolutional neural network using double image features,” in *Proc. DCASE Workshop*, Munich, Germany, 2017, pp. 98–102.
- [75] A. Schindler, T. Lidy, and A. Rauber, “Multi-temporal resolution convolutional neural networks for acoustic scene classification,” in *Proc. DCASE Workshop*, Munich, Germany, 2017, pp. 118–122.
- [76] A. Vafeiadis, D. Kalatzis, K. Votis, D. Giakoumis, D. Tzovaras, L. Chen, and R. Hamzaoui, “Acoustic scene classification: From a hybrid classifier to deep learning,” in *Proc. DCASE Workshop*, Munich, Germany, 2017, pp. 123–127.
- [77] W. Zheng, J. Yi, X. Xing, X. Liu, and S. Peng, “Acoustic scene classification using deep convolutional neural network and multiple spectrograms fusion,” in *Proc. DCASE Workshop*, Munich, Germany, 2017, pp. 133–137.
- [78] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing (3rd Edition)*. London, UK: Pearson Education Ltd., 2014.
- [79] N. De Bruijn, “Uncertainty principles in Fourier analysis,” in *Inequalities (Proc. Sympos. Wright-Patterson Air Force Base, Ohio, 1965)*. Academic Press, New York, NY, USA, 1967, pp. 57–71.
- [80] C. Janott, M. Schmitt, Y. Zhang, K. Qian, V. Pandit, Z. Zhang, C. Heiser, W. Hohenhorst, M. Herzog, W. Hemmert, and B. Schuller, “Snoring classified: The munich passau snore sound corpus,” *Computers in Biology and Medicine*, vol. 94, pp. 106–118, 2018.
- [81] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [82] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, “Extreme learning machine for regression and multiclass classification,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2012.
- [83] Z. Zhang and B. Schuller, “Active learning by sparse instance tracking and classifier confidence in acoustic emotion recognition,” in *Proc. INTER-SPEECH*, Portland, OR, USA, 2012, pp. 362–365.
- [84] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way (3rd Edition)*. Burlington, MA, USA: Elsevier Inc., 2009.
- [85] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley, “Detection and classification of acoustic scenes and events: An IEEE AASP challenge,” in *Proc. WASPAA*, New Paltz, NY, US, 2013, pp. 1–4.

-
- [86] A. Mesaros, T. Heittola, and T. Virtanen, “TUT database for acoustic scene classification and sound event detection,” in *Proc. EUSIPCO*, Budapest, Hungary, 2016, pp. 1128–1132.
- [87] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” in *Proc. NIPS Deep Learning and Representation Learning Workshop*, Montreal, Canada, 2014, pp. 1–9.
- [88] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.
- [89] Z. Zhang, *Semi-Autonomous Data Enrichment and Optimisation for Intelligent Speech Analysis*. Munich, Germany: Technical University of Munich, 2015, Doctoral Thesis.
- [90] J. Benesty, S. Makino, and J. Chen, *Speech Enhancement*. Berlin & Heidelberg, Germany: Springer-Verlag Berlin Heidelberg, 2005.
- [91] A. J. Ferreira, “Combined spectral envelope normalization and subtraction of sinusoidal components in the odft and mdct frequency domains,” in *Proc. WASPAA*, New Paltz, NY, USA, 2001, pp. 51–54.
- [92] P. A. Naylor and N. D. Gaubitch, *Speech Dereverberation*. London, UK: Springer-Verlag London, 2010.
- [93] K. Qian, C. Janott, Z. Zhang, J. Deng, A. Baird, C. Heiser, W. Hohenhorst, M. Herzog, W. Hemmert, and B. Schuller, “Teaching machines on snoring: A benchmark on computer audition for snore sound excitation localisation,” *Archives of Acoustics*, pp. 1–17, 2018, in press.
- [94] G. Fant, *Acoustic Theory of Speech Production: With Calculations based on X-ray Studies of Russian Articulations*. The Hague, Netherlands: De Gruyter Mouton, 1970.
- [95] A. K. Ng, T. San Koh, E. Baey, T. H. Lee, U. R. Abeyratne, and K. Puvanendran, “Could formant frequencies of snore signals be an alternative means for the diagnosis of obstructive sleep apnea?” *Sleep Medicine*, vol. 9, no. 8, pp. 894–898, 2008.
- [96] A. K. Ng, T. San Koh, E. Baey, and K. Puvanendran, “Role of upper airway dimensions in snore production: acoustical and perceptual findings,” *Annals of Biomedical Engineering*, vol. 37, no. 9, pp. 1807–1817, 2009.
- [97] A. Yadollahi and Z. Moussavi, “Formant analysis of breath and snore sounds,” in *Proc. EMBC*, Minneapolis, MN, USA, 2009, pp. 2563–2566.

- [98] T. Murry and R. C. Bone, “Acoustic characteristics of speech following uvulopalatopharyngoplasty,” *The Laryngoscope*, vol. 99, no. 12, pp. 1217–1219, 1989.
- [99] S. McCandless, “An algorithm for automatic formant extraction using linear prediction spectra,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 22, no. 2, pp. 135–141, 1974.
- [100] R. C. Snell and F. Milinazzo, “Formant location from LPC analysis data,” *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 2, pp. 129–134, 1993.
- [101] J. R. Deller Jr, J. H. L. Hansen, and J. G. Proakis, *Discrete Time Processing of Speech Signals*. New York, NY, USA: Wiley-IEEE Press, 1999.
- [102] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control (5th Edition)*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2015.
- [103] S. Agrawal, P. Stone, K. McGuinness, J. Morris, and A. Camilleri, “Sound frequency analysis and the site of snoring in natural and induced sleep,” *Clinical Otolaryngology & Allied Sciences*, vol. 27, no. 3, pp. 162–166, 2002.
- [104] A. S. Karunajeewa, U. R. Abeyratne, and C. Hukins, “Multi-feature snore sound analysis in obstructive sleep apnea–hypopnea syndrome,” *Physiological Measurement*, vol. 32, no. 1, pp. 83–97, 2011.
- [105] M. Cavusoglu, M. Kamasak, O. Erogul, T. Ciloglu, Y. Serinagaoglu, and T. Akcam, “An efficient method for snore/nonsnore classification of sleep sounds,” *Physiological Measurement*, vol. 28, no. 8, pp. 841–853, 2007.
- [106] A. Azarbarzin and Z. Moussavi, “Automatic and unsupervised snore sound extraction from respiratory sound signals,” *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 5, pp. 1156–1162, 2011.
- [107] D. O’Shaughnessy, *Speech Communication: Human and Machine*. New York, NY, USA: Addison-Wesley, 1987.
- [108] B. Schuller, S. Steidl, A. Batliner, A. Vinciarelli, K. Scherer, F. Ringeval, M. Chetouani, F. Weninger, F. Eyben, E. Marchi, M. Mortillaro, H. Salamin, A. Polychroniou, F. Valente, and S. Kim, “The INTERSPEECH 2013 computational paralinguistics challenge: Social signals, conflict, emotion, autism,” in *Proc. INTERSPEECH*, Lyon, France, 2013, pp. 148–152.

-
- [109] F. Eyben, M. Wöllmer, and B. Schuller, “openSMILE—the Munich versatile and fast open-source audio feature extractor,” in *Proc. ACM MM*, Florence, Italy, 2010, pp. 1459–1462.
- [110] F. Eyben, F. Weninger, F. Gross, and B. Schuller, “Recent developments in openSMILE, the Munich open-source multimedia feature extractor,” in *Proc. ACM MM*, Barcelona, Spain, 2013, pp. 835–838.
- [111] B. Schuller, S. Steidl, A. Batliner, E. Bergelson, J. Krajewski, C. Janott, A. Amatuni, M. Casillas, A. Seidl, M. Soderstrom, S. A. Warlaumont, G. Hidalgo, S. Schnieder, C. Heiser, W. Hohenhorst, M. Herzog, M. Schmitt, K. Qian, Y. Zhang, G. Trigeorgis, P. Tzirakis, and S. Zafeiriou, “The INTERSPEECH 2017 computational paralinguistics challenge: Addressee, cold & snoring,” in *Proc. INTERSPEECH*, Stockholm, Sweden, 2017, pp. 3442–3446.
- [112] K. Qian, Z. Zhang, A. Baird, and B. Schuller, “Active learning for bird sound classification via a kernel-based extreme learning machine,” *The Journal of the Acoustical Society of America*, vol. 142, no. 4, pp. 1796–1804, 2017.
- [113] K. Qian, Z. Ren, V. Pandit, Z. Yang, Z. Zhang, and B. Schuller, “Wavelets revisited for the classification of acoustic scenes,” in *Proc. DCASE Workshop*, Munich, Germany, 2017, pp. 108–112.
- [114] B. Schuller, S. Steidl, A. Batliner, J. Epps, F. Eyben, F. Ringeval, E. Marchi, and Y. Zhang, “The INTERSPEECH 2014 computational paralinguistics challenge: Cognitive & physical load,” in *Proc. INTERSPEECH*, Singapore, 2014, pp. 427–431.
- [115] B. Schuller, S. Steidl, A. Batliner, S. Hantke, F. Hönig, J. Orozco-Arroyave, Rafael, E. Nöth, Y. Zhang, and W. Felix, “The INTERSPEECH 2015 computational paralinguistics challenge: Nativeness, parkinson’s & eating condition,” in *Proc. INTERSPEECH*, Dresden, Germany, 2015, pp. 478–482.
- [116] B. Schuller, S. Steidl, A. Batliner, J. Hirschberg, J. Burgoon, K., A. Baird, A. Elkins, Y. Zhang, E. Coutinho, and K. Evanini, “The INTERSPEECH 2016 computational paralinguistics challenge: Deception, sincerity & native language,” in *Proc. INTERSPEECH*, San Francisco, USA, 2016, pp. 2001–2005.
- [117] F. Eyben, *Real-time Speech and Music Classification by Large Audio Feature Space Extraction*. Cham, Switzerland: Springer International Publishing, 2015, Doctoral Thesis.

- [118] L. R. Rabiner and R. W. Schafer, *Theory and Application of Digital Speech Processing (1st Edition)*. Upper Saddle River, NJ, USA: Pearson Higher Education, Inc., 2010.
- [119] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1992.
- [120] R. R. Coifman and M. V. Wickerhauser, “Entropy-based algorithms for best basis selection,” *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 713–718, 1992.
- [121] R. R. Coifman, Y. Meyer, S. Quake, and V. Wickerhauser, “Signal processing and compression with wavelet packets,” in *Wavelets and Their Applications*. Dordrecht, Netherlands: Springer Science & Business Media, 1994, pp. 363–379, J. S. Byrnes, J. L. Byrnes, K. A. Hargreaves, and K. Berry, Ed.
- [122] R. N. Khushaba, *Application of Biosignal-driven Intelligent Systems for Multifunction Prosthesis Control*. Sydney, Australia: University of Technology Sydney, 2010, Doctoral Thesis.
- [123] R. N. Khushaba, S. Kodagoda, S. Lal, and G. Dissanayake, “Driver drowsiness classification using fuzzy wavelet-packet-based feature-extraction algorithm,” *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 1, pp. 121–131, 2011.
- [124] D. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge, UK: Cambridge University Press, 2003.
- [125] M. Schmitt, F. Ringeval, and B. Schuller, “At the border of acoustics and linguistics: Bag-of-audio-words for the recognition of emotions in speech,” in *Proc. INTERSPEECH*, San Francisco, CA, USA, 2016, pp. 495–499.
- [126] M. Schmitt and B. W. Schuller, “openXBOW-introducing the passau open-source crossmodal bag-of-words toolkit,” *Journal of Machine Learning Research*, vol. 18, no. 96, pp. 1–5, 2017.
- [127] Z. S. Harris, “Distributional structure,” *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.
- [128] F. Weninger, P. Staudt, and B. Schuller, “Words that fascinate the listener: Predicting affective ratings of on-line lectures,” *International Journal of Distance Education Technologies*, vol. 11, no. 2, pp. 110–123, 2013.
- [129] J. Sivic and A. Zisserman, “Efficient visual search of videos cast as text retrieval,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 591–606, 2009.

-
- [130] J. Wu, W.-C. Tan, and J. M. Rehg, “Efficient and effective visual codebook generation using additive kernels,” *Journal of Machine Learning Research*, vol. 12, no. Nov, pp. 3097–3118, 2011.
- [131] S. Rawat, P. F. Schulam, S. Burger, D. Ding, Y. Wang, and F. Metze, “Robust audio-codebooks for large-scale event detection in consumer videos,” in *Proc. INTERSPEECH*, Lyon, France, 2013, pp. 2929–2933.
- [132] A. Plinge, R. Grzeszick, and G. A. Fink, “A bag-of-features approach to acoustic event detection,” in *Proc. ICASSP*, Florence, Italy, 2014, pp. 3704–3708.
- [133] H. Lim, M. J. Kim, and H. Kim, “Robust sound event classification using lbp-hog based bag-of-audio-words feature representation,” in *Proc. INTERSPEECH*, Dresden, Germany, 2015, pp. 3325–3329.
- [134] R. Grzeszick, A. Plinge, and G. A. Fink, “Temporal acoustic words for online acoustic event detection,” in *Proc. GCPR*, Aachen, Germany, 2015, pp. 142–153.
- [135] S. Pancoast and M. Akbacak, “Bag-of-audio-words approach for multimedia event classification,” in *Proc. INTERSPEECH*, Portland, OR, USA, 2012, pp. 2105–2108.
- [136] S. Pancoast and M. Akbacak, “N-gram extension for bag-of-audio-words,” in *Proc. ICASSP*, Vancouver, Canada, 2013, pp. 778–782.
- [137] S. Pancoast and M. Akbacak, “Softening quantization in bag-of-audio-words,” in *Proc. ICASSP*, Florence, Italy, 2014, pp. 1370–1374.
- [138] F. B. Pokorny, F. Graf, F. Pernkopf, and B. W. Schuller, “Detection of negative emotions in speech signals using bags-of-audio-words,” in *Proc. ACII*, Xi’an, P. R. China, 2015, pp. 879–884.
- [139] J. Joshi, R. Goecke, S. Alghowinem, A. Dhall, M. Wagner, J. Epps, G. Parker, and M. Breakspear, “Multimodal assistive technologies for depression diagnosis and monitoring,” *Journal on Multimodal User Interfaces*, vol. 7, no. 3, pp. 217–228, 2013.
- [140] S. Rawat, P. F. Schulam, S. Burger, D. Ding, Y. Wang, and F. Metze, “Robust audio-codebooks for large-scale event detection in consumer videos,” in *Proc. INTERSPEECH*. Lyon, France: ISCA, 2013, pp. 2929–2933.
- [141] D. Arthur and S. Vassilvitskii, “K-means++: The advantages of careful seeding,” in *Proc. ACM-SIAM SODA*, New Orleans, LA, USA, 2007, pp. 1027–1035.

- [142] J. Sola and J. Sevilla, “Importance of input data normalization for the application of neural networks to complex industrial problems,” *IEEE Transactions on Nuclear Science*, vol. 44, no. 3, pp. 1464–1468, 1997.
- [143] M. N. Murty and V. S. Devi, *Pattern Recognition: An Algorithmic Approach*. Dordrecht, Netherlands: Springer Science & Business Media, 2011.
- [144] H. Pishro-Nik, *Introduction to Probability, Statistics, and Random Processes*. Electrical and Computer Engineering Educational Materials, 2014, http://scholarworks.umass.edu/ece_ed_materials/1.
- [145] G. H. John and P. Langley, “Estimating continuous distributions in bayesian classifiers,” in *Proc. UAI*, Montreal, Canada, 1995, pp. 338–345.
- [146] A. McCallum, K. Nigam *et al.*, “A comparison of event models for naive bayes text classification,” in *Proc. AAAI Workshop on Learning for Text Categorization*, Madison, WI, USA, 1998, pp. 41–48.
- [147] V. Metsis, I. Androutsopoulos, and G. Paliouras, “Spam filtering with naive bayes-which naive bayes?” in *Proc. CEAS*, Mountain View, CA, USA, 2006, pp. 1–9.
- [148] H. Zhang, “The optimality of naive bayes,” in *Proc. FLAIRS*, Miami Beach, FL, USA, 2004, pp. 562–567.
- [149] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [150] M. Deza and E. Deza, *Encyclopedia of Distances*. Berlin & Heidelberg, Germany: Springer-Verlag Berlin Heidelberg, 2013.
- [151] J. C. Platt, “Fast training of support vector machines using sequential minimal optimization,” in *Advances in Kernel Methods: Support Vector Learning*. Cambridge, MA, USA: MIT Press, 1999, pp. 185–208.
- [152] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.
- [153] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [154] J. R. Quinlan, “Induction of decision trees,” *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

-
- [155] D. W. Opitz and R. Maclin, “Popular ensemble methods: An empirical study,” *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–198, 1999.
- [156] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [157] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [158] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” in *Proc. NIPS*, Vancouver, Canada, 2006, pp. 153–160.
- [159] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.
- [160] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. NIPS*, Stateline, NV, USA, 2012, pp. 1097–1105.
- [161] C. Szegedy, A. Toshev, and D. Erhan, “Deep neural networks for object detection,” in *Proc. NIPS*, Stateline, NV, USA, 2013, pp. 2553–2561.
- [162] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, “Handwritten digit recognition with a back-propagation network,” in *Proc. NIPS*, Denver, CO, USA, 1989, pp. 396–404.
- [163] S. Amiriparian, M. Gerczuk, S. Ottl, N. Cummins, M. Freitag, S. Pugachevskiy, A. Baird, and B. Schuller, “Snore sound classification using image-based deep spectrum features,” in *Proc. INTERSPEECH*, Stockholm, Sweden, 2017, pp. 3512–3516.
- [164] M. Freitag, S. Amiriparian, N. Cummins, M. Gerczuk, and B. Schuller, “An end-to-evolutionhybrid approach for snore sound classification,” in *Proc. INTERSPEECH*, Stockholm, Sweden, 2017, pp. 3507–3511.
- [165] Z. Ren, V. Pandit, K. Qian, Z. Yang, Z. Zhang, and B. Schuller, “Deep sequential image features on acoustic scene classification,” in *Proc. DCASE Workshop*, Munich, Germany, 2017, pp. 113–117.
- [166] Z. Ren, K. Qian, Z. Zhang, V. Pandit, A. Baird, and B. Schuller, “Deep scalogram representations for acoustic scene classification,” *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 3, pp. 662–669, 2018.

- [167] C. M. Bishop, *Pattern recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
- [168] C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia, “Incorporating second-order functional knowledge for better option pricing,” in *Proc. NIPS*, Vancouver, Canada, 2001, pp. 472–478.
- [169] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, “Maxout networks,” in *Proc. ICML*, Atlanta, GA, USA, 2013, pp. 1319–1327.
- [170] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, p. 533, 1986.
- [171] P. J. Werbos, “Backpropagation through time: what it does and how to do it,” *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.
- [172] R. A. Jacobs, “Increased rates of convergence through learning rate adaptation,” *Neural Networks*, vol. 1, no. 4, pp. 295–307, 1988.
- [173] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in *Proc. COMPSTAT*, Paris, France, 2010, pp. 177–186.
- [174] B. T. Polyak, “Some methods of speeding up the convergence of iteration methods,” *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.
- [175] Y. Nesterov, “A method of solving a convex programming problem with convergence rate $\mathcal{O}(1/\sqrt{k})$,” vol. 27, no. 2, pp. 372–376, 1983.
- [176] M. F. Møller, “A scaled conjugate gradient algorithm for fast supervised learning,” *Neural Networks*, vol. 6, no. 4, pp. 525–533, 1993.
- [177] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, USA: Springer, 2006.
- [178] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [179] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. ICML*, Lille, France, 2015, pp. 448–456.
- [180] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

-
- [181] H. Bourlard and Y. Kamp, “Auto-association by multilayer perceptrons and singular value decomposition,” *Biological Cybernetics*, vol. 59, no. 4-5, pp. 291–294, 1988.
- [182] G. E. Hinton and R. S. Zemel, “Autoencoders, minimum description length and helmholtz free energy,” in *Proc. NIPS*, Denver, CO, USA, 1993, pp. 3–10.
- [183] B. A. Olshausen and D. J. Field, “Sparse coding with an overcomplete basis set: A strategy employed by V1?” *Vision Research*, vol. 37, no. 23, pp. 3311–3325, 1997.
- [184] H. Lee, C. Ekanadham, and A. Y. Ng, “Sparse deep belief net model for visual area V2,” in *Proc. NIPS*, Vancouver, Canada, 2008, pp. 873–880.
- [185] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
- [186] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *Proc. ICML*, Helsinki, Finland, 2008, pp. 1096–1103.
- [187] N. Morgan and H. Bourlard, “Continuous speech recognition using multilayer perceptrons with hidden Markov models,” in *Proc. ICASSP*, Albuquerque, NM, USA, 1990, pp. 413–416.
- [188] O. Vinyals, S. V. Ravuri, and D. Povey, “Revisiting recurrent neural networks for robust ASR,” in *Proc. ICASSP*, Kyoto, Japan, 2012, pp. 4085–4088.
- [189] J. L. Elman, “Finding structure in time,” *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.
- [190] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, “Gradient flow in recurrent nets: The difficulty of learning long-term dependencies,” in *A Field Guide to Dynamical Recurrent Neural Networks*. Piscataway, NJ, USA: IEEE Press, 2001, pp. 237–244, J. F. Kolen, and S. C. Kremer, Ed.
- [191] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *Proc. ICML*, Atlanta, GA, USA, 2013, pp. 1310–1318.
- [192] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [193] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, “Extreme learning machine: a new learning scheme of feedforward neural networks,” in *Proc. IJCNN*, Budapest, Hungary, 2004, pp. 985–990.

- [194] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, “Extreme learning machine: Theory and applications,” *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.
- [195] G. Huang, G.-B. Huang, S. Song, and K. You, “Trends in extreme learning machines: A review,” *Neural Networks*, vol. 61, pp. 32–48, 2015.
- [196] G.-B. Huang, L. Chen, C. K. Siew *et al.*, “Universal approximation using incremental constructive feedforward networks with random hidden nodes,” *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.
- [197] G.-B. Huang and L. Chen, “Convex incremental extreme learning machine,” *Neurocomputing*, vol. 70, no. 16-18, pp. 3056–3062, 2007.
- [198] G.-B. Huang and L. Chen, “Enhanced random search based incremental extreme learning machine,” *Neurocomputing*, vol. 71, no. 16-18, pp. 3460–3468, 2008.
- [199] G.-B. Huang, “An insight into extreme learning machines: random neurons, random features and kernels,” *Cognitive Computation*, vol. 6, no. 3, pp. 376–390, 2014.
- [200] G.-B. Huang, “What are extreme learning machines? Filling the gap between Frank Rosenblatt’s dream and John von Neumann’s puzzle,” *Cognitive Computation*, vol. 7, no. 3, pp. 263–278, 2015.
- [201] D. Serre, *Matrices: Theory and Applications*. New York, NY, USA: Springer-Verlag New York, Inc., 2002.
- [202] C. R. Rao and S. K. Mitra, *Generalized Inverse of Matrices and Its Applications*. New York, NY, USA: Wiley, 2002.
- [203] X. Zhu and A. B. Goldberg, “Introduction to semi-supervised learning,” *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 3, no. 1, pp. 1–130, 2009.
- [204] B. Settles, “Active learning literature survey,” University of Wisconsin–Madison, Madison, WI, USA, Computer Sciences Technical Report 1648, 2009.
- [205] D. D. Lewis and W. A. Gale, “A sequential algorithm for training text classifiers,” in *Proc. ACM SIGIR*, Dublin, Ireland, 1994, pp. 3–12.
- [206] T. Scheffer, C. Decomain, and S. Wrobel, “Active hidden markov models for information extraction,” in *Proc. IDA*, Cascais, Portugal, 2001, pp. 309–318.
- [207] J. Platt, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” *Advances in Large Margin Classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

-
- [208] T. Hastie and R. Tibshirani, “Classification by pairwise coupling,” in *Proc. NIPS*, Denver, CO, USA, 1997, pp. 507–513.
- [209] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: an update,” *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.
- [210] T.-F. Wu, C.-J. Lin, and R. C. Weng, “Probability estimates for multi-class classification by pairwise coupling,” *Journal of Machine Learning Research*, vol. 5, no. Aug, pp. 975–1005, 2004.
- [211] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [212] T. G. Dietterich, “Approximate statistical tests for comparing supervised classification learning algorithms,” *Neural Computation*, vol. 10, no. 7, pp. 1895–1923, 1998.
- [213] M. R. Spiegel, J. J. Schiller, R. A. Srinivasan, and M. LeVan, *Probability and Statistics*. New York, NY, USA: McGraw-Hill, 2009.
- [214] T. Kendzerska, A. S. Gershon, G. Hawker, G. Tomlinson, and R. S. Leung, “Obstructive sleep apnea and incident diabetes. a historical cohort study,” *American Journal of Respiratory and Critical Care Medicine*, vol. 190, no. 2, pp. 218–225, 2014.
- [215] B. Mokhlesi, S. Ham, and D. Gozal, “The effect of sex and age on the comorbidity burden of osa: an observational analysis from a large nationwide us health claims database,” *The European Respiratory Journal*, vol. 47, no. 4, pp. 1162–1169, 2016.
- [216] P. E. Peppard, M. Szklo-Coxe, K. M. Hla, and T. Young, “Longitudinal association of sleep-related breathing disorder and depression,” *Archives of Internal Medicine*, vol. 166, no. 16, pp. 1709–1715, 2006.
- [217] D. Leger, V. Bayon, J. P. Laaban, and P. Philip, “Impact of sleep apnea on economics,” *Sleep Medicine Reviews*, vol. 16, no. 5, pp. 455–462, 2012.
- [218] M. S. Aldrich, *Sleep Medicine*. New York, NY, USA: Oxford University Press, 1999.
- [219] A. Roebuck, V. Monasterio, E. Gederri, M. Osipov, J. Behar, A. Malhotra, T. Penzel, and G. Clifford, “A review of signals used in sleep analysis,” *Physiological Measurement*, vol. 35, no. 1, pp. R1–R57, 2014.

- [220] M. Reda, G. J. Gibson, and J. A. Wilson, “Pharyngoesophageal pressure monitoring in sleep apnea syndrome,” *Otolaryngology–Head and Neck Surgery*, vol. 125, no. 4, pp. 324–331, 2001.
- [221] H. Demin, Y. Jingying, W. J. Y. Qingwen, L. Yuhua, and W. Jiangyong, “Determining the site of airway obstruction in obstructive sleep apnea with airway pressure measurements during sleep,” *The Laryngoscope*, vol. 112, no. 11, pp. 2081–2085, 2002.
- [222] B. A. Stuck and J. T. Maurer, “Airway evaluation in obstructive sleep apnea,” *Sleep Medicine Reviews*, vol. 12, no. 6, pp. 411–436, 2008.
- [223] S. Miyazaki, Y. Itasaka, K. Ishikawa, and K. Togawa, “Acoustic analysis of snoring and the site of airway obstruction in sleep related respiratory disorders,” *Acta Oto-Laryngologica*, vol. 118, no. 537, pp. 47–51, 1998.
- [224] P. Hill, B. Lee, J. Osborne, and E. Osman, “Palatal snoring identified by acoustic crest factor analysis,” *Physiological Measurement*, vol. 20, no. 2, pp. 167–174, 1999.
- [225] R. J. Beeton, I. Wells, P. Ebden, H. Whittet, and J. Clarke, “Snore site discrimination using statistical moments of free field snoring sounds recorded during sleep nasendoscopy,” *Physiological Measurement*, vol. 28, no. 10, pp. 1225–1236, 2007.
- [226] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models*. Heidelberg, Germany: Springer Verlag, 1999, 2nd Edition.
- [227] M. Herzog, S. Plöfl, A. Glien, B. Herzog, C. Rohrmeier, T. Kühnel, S. Plontke, and P. Kellner, “Evaluation of acoustic characteristics of snoring sounds obtained during drug-induced sleep endoscopy,” *Sleep and Breathing*, vol. 3, no. 19, pp. 1011–1019, 2014.
- [228] E. J. Kezirian, W. Hohenhorst, and N. de Vries, “Drug-induced sleep endoscopy: the vote classification,” *European Archives of Oto-Rhino-Laryngology*, vol. 268, no. 8, pp. 1233–1236, 2011.
- [229] N. Hessel and N. de Vries, “Diagnostic work-up of socially unacceptable snoring,” *European Archives of Oto-Rhino-Laryngology*, vol. 259, no. 3, pp. 158–161, 2002.
- [230] H. Kaya and K. A. Alexey, “Introducing weighted kernel classifiers for handling imbalanced paralinguistic corpora: Snoring, addressee and cold,” in *Proc. INTERSPEECH*, Stockholm, Sweden, 2017, pp. 3527–3531.

-
- [231] G. Gosztolya, R. Busa-Fekete, T. Grósz, and L. Tóth, “DNN-based feature extraction and classifier combination for child-directed speech, cold and snoring identification,” in *Proc. INTERSPEECH*, Stockholm, Sweden, 2017, pp. 3522–3526.
- [232] S. Rännar, F. Lindgren, P. Geladi, and S. Wold, “A PLS kernel algorithm for data sets with many variables and fewer objects. part 1: Theory and algorithm,” *Journal of Chemometrics*, vol. 8, no. 2, pp. 111–125, 1994.
- [233] W. Zong, G.-B. Huang, and Y. Chen, “Weighted extreme learning machine for imbalance learning,” *Neurocomputing*, vol. 101, pp. 229–242, 2013.
- [234] K. Li, X. Kong, Z. Lu, L. Wenyin, and J. Yin, “Boosting weighted elm for imbalanced learning,” *Neurocomputing*, vol. 128, pp. 15–21, 2014.
- [235] H. Kaya, A. A. Karpov, and A. A. Salah, “Fisher vectors with cascaded normalization for paralinguistic analysis,” in *Proc. INTERSPEECH*, Dresden, Germany, 2015, pp. 909–913.
- [236] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. ICLR*, San Diego, CA, USA, 2015, pp. 1–14.
- [237] R. Cheng and Y. Jin, “A competitive swarm optimizer for large scale optimization,” *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 191–204, 2015.
- [238] S. Gu, R. Cheng, and Y. Jin, “Feature selection for high-dimensional classification using a competitive swarm optimizer,” *Soft Computing*, vol. 22, no. 3, pp. 811–822, 2018.
- [239] C. K. Catchpole and P. J. Slater, *Bird Song: Biological Themes and Variations*. Cambridge, UK: Cambridge University Press, 2003.
- [240] K. Kaewtip, A. Alwan, C. O’Reilly, and C. E. Taylor, “A robust automatic birdsong phrase classification: A template-based approach,” *The Journal of the Acoustical Society of America*, vol. 140, no. 5, pp. 3691–3701, 2016.
- [241] H. Goëau, H. Glotin, W.-P. Vellinga, R. Planqué, A. Rauber, and A. Joly, “Lifeclef bird identification task 2014,” in *CLEF Working Notes*, Sheffield, UK, 2014, pp. 585–597.
- [242] A. Sevilla and G. H., “Audio bird classification with inception-v4 extended with time and time-frequency attention mechanisms,” in *CLEF Working Notes*, Dublin, Ireland, 2017, pp. 1–8.

- [243] B. Fazekas, A. Schindler, T. Lidy, and A. Rauber, “A multi-modal deep neural network approach to bird-song identification,” in *CLEF Working Notes*, Dublin, Ireland, 2017, pp. 1–6.
- [244] S. Kahl, T. Wilhelm-Stein, H. Hussein, H. Klinck, D. Kowerko, M. Ritter, and M. Eibl, “Large-scale bird sound classification using convolutional neural networks,” in *CLEF Working Notes*, Dublin, Ireland, 2017, pp. 1–14.
- [245] K. Qian, Z. Zhang, A. Baird, and B. Schuller, “Active learning for bird sounds classification,” *Acta Acustica united with Acustica*, vol. 103, no. 3, pp. 361–364, 2017.
- [246] K. Qian, Z. Zhang, F. Ringeval, and B. Schuller, “Bird sounds classification by large scale acoustic features and extreme learning machine,” in *Proc. GlobalSIP*, Orlando, FL, USA, 2015, pp. 1317–1321.
- [247] E. G. Horta and A. de Pádua Braga, “An extreme learning approach to active learning,” in *Proc. ESANN*, Bruges, Belgium, 2014, pp. 613–618.
- [248] H. Yu, C. Sun, W. Yang, X. Yang, and X. Zuo, “AL-ELM: One uncertainty-based active learning algorithm using extreme learning machine,” *Neurocomputing*, vol. 166, pp. 140–150, 2015.
- [249] Y. Zhang and M. J. Er, “Sequential active learning using meta-cognitive extreme learning machine,” *Neurocomputing*, vol. 173, pp. 835–844, 2016.
- [250] B. Schölkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT press, 2002.
- [251] D. Li, J. Tam, and D. Toub, “Auditory scene classification using machine learning techniques,” in *Proc. DCASE*, 2013, pp. 1–3.
- [252] T. Tieleman and G. Hinton, “RMSprop: Divide the gradient by a running average of its recent magnitude,” in *COURSERA: Neural Networks for Machine Learning, Lecture 6.5*, 2012.
- [253] S. V. Vaseghi, *Advanced Digital Signal Processing and Noise Reduction*. Chichester, UK: John Wiley & Sons, Inc., 2008.
- [254] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Proc. NIPS*, Montreal, Canada, 2014, pp. 2672–2680.

- [255] S. Mun, S. Park, D. K. Han, and H. Ko, “Generative adversarial network based acoustic scene training set augmentation and selection using SVM hyperplane,” in *Proc. DCASE Workshop*, Munich, Germany, 2017, pp. 93–97.
- [256] J. D. Cutnell and K. W. Johnson, *Physics (4th Edition)*. New York, NY, USA: John Wiley & Sons, Inc., 1997.
- [257] L. T. Nwe, D. H. Tran, T. Z. W. Ng, and B. Ma, “An integrated solution for snoring sound classification using bhattacharyya distance based GMM super-vectors with SVM, feature selection with random forest and spectrogram with CNN,” in *Proc. INTERSPEECH*, Stockholm, Sweden, 2017, pp. 3467–3471.
- [258] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio Set: An ontology and human-labeled dataset for audio events,” in *Proc. ICASSP*, New Orleans, LA, USA, 2017, pp. 776–780.
- [259] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [260] R. Caruana, *Multitask Learning*. Pittsburgh, PA, USA: Carnegie Mellon University, 1997, Doctoral Thesis.
- [261] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*. Cambridge, MA, USA: MIT Press, 2006.
- [262] Z. Zhang, E. Coutinho, J. Deng, and B. Schuller, “Cooperative learning and its application to emotion recognition from speech,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 115–126, 2015.