TECHNISCHE UNIVERSITÄT MÜNCHEN
Lehrstuhl für Mensch-Maschine-Kommunikation

Deep Representation Learning Techniques for Audio Signal Processing

Shahin Amiriparian

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. Eckehard Steinbach
Prüfer der Dissertation:	1. Prof. Dr.-Ing. habil. Björn W. Schuller
	2. Prof. Dr.-Ing. Sami Haddadin

Die Dissertation wurde am 28.11.2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 09.12.2019 angenommen.

Acknowledgements

First of all, I would like to express my gratitude to my supervisor, Professor Björn Schuller, for his excellent guidance and inspirational supervision. I have always benefited from his helpful comments regarding my scientific thinking and writing. His professional and personal advice guided me throughout my time as his doctoral student. I also want to thank my mentor, Dr. Martin Heckmann, for the supportive talks and discussions. Furthermore, I would like to thank my dear colleagues for productive collaborations and the nice time at the university. Especially, I want to thank Alice Baird and Maximilian Schmitt for their valuable comments on this thesis. Special thanks also go to the co-authors of the pre-publications being collected in this thesis, including Michael Freitag, Maurice Gerczuk, Sandra Ottl, and Sergey Pugachevskiy. I want to sincerely thank Magda for all her support during the hardest time of my doctoral study. Heartfelt thanks go to Ingrid Gerlach for her support and the great and inspirational talks we had with each other. I would like to thank Professor Wolfgang Günther for his advice and guidance. I want to express my special thanks to Anna for all her support and encouragement during the time of writing this thesis. Last but not least, I heartily thank my sister Shahla, my brothers Afshin and Ramin, and my parents Farideh and Hassan, for their unconditional love and continuous support.

Abstract

This thesis investigates the potential of deep neural networks for representations learning from audio signals. First, it is shown that visual representations of audio signals, such as spectrograms, consist of a variety of meaningful acoustic information due to the richness of their time-frequency information. Second, it is demonstrated that state-of-the-art deep neural network-based methodologies are highly suitable for the task of representation learning from the generated spectra. In this regard, novel learning models based on convolutional and recurrent neural networks are proposed. Furthermore, extensive experiments are conducted to evaluate the practicability of the introduced techniques for a wide range of audio recognition tasks, including analysis of acoustic environments, speech processing, and health monitoring. Finally, it is shown that the developed systems are able to learn meaningful and robust representations.

Preface

This dissertation is submitted for the degree of Doctor of Engineering (Dr.-Ing.) at the Technische Universität München (TUM). The research described herein is based on selected pre-publications made during my time as one of Professor Björn Schuller's doctoral students. The selection is based upon scientific relevance and novelty, with the intention to cover a range of deep learning experiments for the task of representation learning from audio. The first aim of this dissertation is to grant these pre-publications a broader access for a knowledgeable reader, who is already acquainted with the basics of machine learning and audio signal processing. The second aim is to provide a wider view of potential fields of application of deep representation learning, as described throughout the pre-publications. For this purpose, the pre-publications have been broken down and restructured into "Data and Procedure" and "Results", put into a general research framework, and enhanced by an introductory chapter and concluding remarks. It is hoped that this results in a publication that one enjoys reading from the beginning to the end. Proceeding this, the interested reader can find the full list of my pre-publications.

Augsburg, Fall 2018 Shahin Amiriparian

List of Publications

1. **S. Amiriparian**, N. Cummins, M. Gerczuk, S. Pugachevskiy, S. Ottl, and B. Schuller,""Are you playing a shooter again?!" deep representation learning for audio-based video game genre recognition," *IEEE Transactions on Games*, 2018, 11 pages, to appear.

2. **S. Amiriparian**, A. Baird, S. Julka, A. Alcorn, S. Ottl, S. Petrović, E. Ainger, N. Cummins, and B. Schuller, "Recognition of Echolalic Autistic Child Vocalisations Utilising Convolutional Recurrent Neural Networks," in *Proceedings of INTERSPEECH 2018, 19th Annual Conference of the International Speech Communication Association*. Hyderabad, India: ISCA, September 2018, pp. 2334– 2338.

3. **S. Amiriparian**, M. Freitag, N. Cummins, M. Gerzcuk, S. Pugachevskiy, and B. Schuller, "A Fusion of Deep Convolutional Generative Adversarial Networks and Sequence to Sequence Autoencoders for Acoustic Scene Classification," in *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)*, EURASIP. Rome, Italy: IEEE, September 2018, pp. 982–986.

4. **S. Amiriparian**, M. Schmitt, N. Cummins, K. Qian, F. Dong, and B. Schuller, "Deep Unsupervised Representation Learning for Abnormal Heart Sound Classification," in *Proceedings of the 40th Annual International Conference of the IEEE Engineering in Medicine & Biology Society, EMBC 2018*, IEEE. Honolulu, HI: IEEE, July 2018, pp. 4776–4779.

5. **S. Amiriparian**, M. Gerczuk, S. Ottl, N. Cummins, S. Pugachevskiy, and B. Schuller, "Bag-of-deep-features: Noise-robust deep feature representations for audio analysis," in *Proceedings of the 31st International Joint Conference on Neural Networks (IJCNN)*. Rio de Janeiro, Brazil: IEEE, July 2018, pp. 2419–2425.

6. **S. Amiriparian**, M. Schmitt, S. Hantke, V. Pandit, and B. Schuller, "Humans Inside: Cooperative Big Multimedia Data Mining," in *Innovations in Big Data Mining and Embedded Knowledge: Domestic and Social Context Challenges*. ser. Intelligent Systems Reference Library (ISRL), A. Esposito, A. M. Esposito, and L. C. Jain, Eds. Springer, 2018, 25 pages, to appear.

7. **S. Amiriparian**, S. Julka, N. Cummins, and B. Schuller, "Deep Convolutional Recurrent Neural Networks for Rare Sound Event Detection," in *Proceedings of 44. Jahrestagung für Akustik (DAGA)*, Munich, Germany, March 2018, pp. 1522–1525.

8. **S. Amiriparian**, M. Freitag, N. Cummins, and B. Schuller, "Sequence To Sequence Autoencoders for Unsupervised Representation Learning From Audio," in *Proceedings of the 2nd Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE 2017)*. Munich, Germany: IEEE, November 2017, pp. 17–21.

9. **S. Amiriparian**, N. Cummins, S. Ottl, M. Gerczuk, and B. Schuller, "Sentiment Analysis Using Image-based Deep Spectrum Features," in *Proceedings of the Biannual Conference on Affective Computing and Intelligent Interaction (ACII)*, San Antonio, TX, 2017, pp. 26–29.

10. **S. Amiriparian**, M. Freitag, N. Cummins, and B. Schuller, "Feature Selection in Multimodal Continuous Emotion Prediction," in *Proceedings of the 2nd International Workshop on Automatic Sentiment Analysis in the Wild (WASA 2017) held in conjunction with the 7th biannual Conference on Affective Computing and Intelligent Interaction (ACII 2017)*, AAAC. San Antonio, TX: IEEE, October 2017, pp. 30–37.

11. **S. Amiriparian**, S. Pugachevskiy, N. Cummins, S. Hantke, J. Pohjalainen, G. Keren, and B. Schuller, "CAST a database: Rapid targeted large-scale big data acquisition via small-world modelling of social media platforms," in *Proceedings of the Biannual Conference on Affective Computing and Intelligent Interaction (ACII)*, San Antonio, TX, 2017, pp. 340–345.

12. **S. Amiriparian**, M. Gerczuk, S. Ottl, N. Cummins, M. Freitag, S. Pugachevskiy, and B. Schuller, "Snore sound classification using image-based deep spectrum features," in *Proceedings of INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association*. Stockholm, Sweden: ISCA, August 2017. pp. 3512–3516.

13. **Nominated for best student paper award:**
S. Amiriparian, J. Pohjalainen, E. Marchi, S. Pugachevskiy, and B. Schuller, "Is deception emotional? An emotion-driven predictive approach," in *Proceedings of INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association*. San Francisco, CA: ISCA, September 2016, pp. 2011–2015.

14. **S. Amiriparian**, N. Cummins, M. Freitag, K. Qian, R. Zhao, V. Pandit and B. Schuller, "The Combined Augsburg / Passau / TUM / ICL System for DCASE 2017," in *Proceedings of the 2nd Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE 2017)*. Munich, Germany: IEEE, November 2017. 1 page. Technical report.

15. M. Freitag, **S. Amiriparian**, S. Pugachevskiy, N. Cummins, and B. Schuller, "auDeep: Unsupervised Learning of Representations from Audio with Deep Recurrent Neural Networks," *Journal of Machine Learning Research*, vol. 18, no. 173, pp. 1–5, 2018.

16. N. Cummins, **S. Amiriparian**, S. Ottl, M. Gerczuk, M. Schmitt, and B. Schuller, "Multimodal Bag-of-Words for Cross Domains Sentiment Analysis," in *Proceedings of the 43rd IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2018*, Calgary, Canada, IEEE, April 2018. pp. 1–5.

17. N. Cummins, **S. Amiriparian**, G. Hagerer, A. Batliner, S. Steidl, and B. Schuller, "An image-based deep spectrum feature representation for the recognition of emotional speech," *in Proceedings of the 25th ACM International Conference on Multimedia, MM 2017*. Mountain View, CA: ACM, October 2017, pp. 478–484.

18. B. Schuller, S. Steidl, A. Batliner, P. B. Marschik, H. Baumeister, F. Dong, S. Hantke, F. Pokorny, E.-M. Rathner, K. D. Bartl-Pokorny, C. Einspieler, D. Zhang, A. Baird, **S. Amiriparian**, K. Qian, Z. Ren, M. Schmitt, P. Tzirakis, and S. Zafeiriou, "The INTERSPEECH 2018 Computational Paralinguistics Challenge: Atypical & Self-Assessed Affect, Crying & Heart Beats," in *Proceedings of INTERSPEECH 2018, 19th Annual Conference of the International Speech Communication Association*. Hyderabad, India: ISCA, September 2018, pp. 122–126.

19. M. Freitag, **S. Amiriparian**, N. Cummins, M. Gerczuk, and B. Schuller, "An 'End-to-Evolution' Hybrid Approach for Snore Sound Classification," in *Proceedings of INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association*. Stockholm, Sweden: ISCA, August 2017, pp. 3507–3511.

20. V. Pandit, **S. Amiriparian**, M. Schmitt, A. Mousa, and B. Schuller, "Big Data Multimedia Mining: Feature Extraction facing Volume, Velocity, and Variety," in *Big Data Analytics for Large-Scale Multimedia Search*, S. Vrochidis, B. Huet, E. Chang, and I. Kompatsiaris, Eds. Wiley, 2017.

21. A. Baird, **S. Amiriparian**, N. Cummins, A. M. Alcorn, A. Batliner, S. Pugachevskiy, M. Freitag, M. Gerczuk, and B. Schuller, "Automatic Classification of Autistic Child Vocalisations: A Novel Database and Results," in *Proceedings of INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association*. Stockholm, Sweden: ISCA, Augsust 2017, pp. 849–853.

22. A. Baird, **S. Amiriparian**, A. Rynkiewicz, and B. Schuller, "Echolalic Autism Spectrum Condition Vocalisations: Brute-Force and Deep Spectrum Features," in *Proceedings of the International Paediatric Conference (IPC 2018)*. Rzeszów, Poland: Polish Society of Social Medicine and Public Health, May 2018, 2 pages, to appear.

23. F. Ringeval, **S. Amiriparian**, F. Eyben, K. Scherer, and B. Schuller, "Emotion Recognition in the Wild: Incorporating Voice and Lip Activity in Multimodal Decision-Level Fusion," in *Proceedings of the ICMI 2014 EmotiW – Emotion Recognition In The Wild Challenge and Workshop (EmotiW 2014), Satellite of the 16th ACM International Conference on Multimodal Interaction (ICMI 2014)*, Istanbul, Turkey, ACM, November 2014, pp. 473–480.

24. F. Ringeval, B. Schuller, M. Valstar, R. Cowie, H. Kaya, M. Schmitt, **S. Amiriparian**, N. Cummins, D. Lalanne, A. Michaud, E. Ciftci, H. Gulec, A. A. Salah, and M. Pantic,

"AVEC 2018 Workshop and Challenge: Bipolar Disorder and Cross-Cultural Affect Recognition," in *Proceedings of the 2018 on Audio/Visual Emotion Challenge and Workshop*, ser. AVEC'18. Seoul, Republic of Korea: ACM, 2018, pp. 3–13.

25. N. Cummins, M. Schmitt, **S. Amiriparian**, J. Krajewski, and B. Schuller, "You sound ill, take the day off: Classification of speech affected by Upper Respiratory Tract Infection," in *Proceedings of the 39th Annual International Conference of the IEEE Engineering in Medicine & Biology Society, EMBC 2017*, Jeju Island, South Korea, IEEE, July 2017, pp. 3806–3809.

26. F. Demir, A. Sengur, N. Cummins, **S. Amiriparian**, and B. Schuller, "Low level texture features for snore sound discrimination," in *Proceedings of the 40th Annual International Conference of the IEEE Engineering in Medicine & Biology Society, EMBC 2018*, Honolulu, HI, IEEE, July 2018. pp. 413-416.

27. A. Sengur, F. Demir, H. Lu, **S. Amiriparian**, N. Cummins, and B. Schuller, "Compact bilinear deep features for environmental sound recognition," in *Proceedings of the International Conference on Artificial Intelligence and Data Mining, IDAP 2018*, Malatya, Turkey, IEEE, September 2018. 5 pages, to appear.

Contributions in **German**:

1. **S. Amiriparian**, M. Schmitt, B. Schuller, "Exploiting Deep Learning: die wichtigsten Bits und Pieces - Sieh zu und lerne", Java Magazin, Ausgabe 5. 2018, S. 46-53, JAXenter, 2018.

2. **S. Amiriparian**, A. Baird, B. Schuller, "Automatische Erkennung von Echolalie bei Kindern mit Autismus-Spektrum-Störung mithilfe direkter Mensch-Roboter-Interaktion", Deutscher Kongress für Psychosomatische Medizin und Psychotherapie, Berlin, Deutschland, März 2019.

3. M. Schmitt, **S. Amiriparian**, B. Schuller, "Maschinelle Sprachverarbeitung: Wie lernen Computer unsere Sprache zu verstehen?", Entwickler Magazin, Spezial Volume 17, S. 30-38, Software & Support Media GmbH, 2018.

Contents

Part I

Introduction

1

Introduction

1.1 Motivation

We are constantly surrounded by dynamic audio events, some pleasant, such as singing birds and babbling brooks, other less so, like the sound of a lawnmower on a Saturday morning. From an early age, humans have the ability to recognise, filter, and understand a rich variety of existing sounds, focusing on important details in the audio whilst channeling out a large number of distractions. In the era of *Artificial Intelligence* (AI), it is vital for machines to analyse and understand the acoustic environment with high precision. In this regard, machine learning systems for detection of acoustic events, e. g. for autonomous-driving cars [1] and acoustic surveillance [2] or the application of *Automatic Speech Recognition* (ASR) and emotion recognition systems for human-computer interaction [3, 4] are gaining in popularity. Likewise, research in the field of mobile health and remote health monitoring based on acoustic information are showing strong improvements [5–7]. For all these applications, there is a need for machine learning systems which have robust performance and high accuracy in real-world conditions. In this regard, this thesis not only investigates recent deep learning techniques for audio processing, but also proposes novel machine learning paradigms, offering a solution for the aforementioned problems.

1.2 Problem Statement

Despite recent rapid developments in the field of machine learning, current computer audition systems are still unable to perform the task of audio understanding and analysis with human-like precision. Furthermore, conventional machine learning methods were restricted in their ability to process the input data in its raw format. For many years, building machine learning systems required precise engineering and fundamental domain knowledge to develop a feature extractor (e. g. OPENSMILE [8, 9]) which mapped characteristics of the raw audio, such as amplitude and frequency, into meaningful feature

vectors from which the machine learner could recognise patterns. This manual process of feature engineering is tedious and time consuming due to the large amount of human intervention needed. Such features are also highly task-dependent, i. e. they require human expertise to select and design discriminative information for a given task [10]. However, task-specific feature sets often show stronger performance in comparison with more general feature vectors. As these features are fine-tuned for a specific domain, e. g. acoustic features for affective computing, classification of emotions [11–13] and for music genre classification [14], they may not perform well for other tasks which differ slightly and can deteriorate the performance of the machine learning system [15].

1.3 Objectives

For the reasons mentioned above, there is a need for manual adjustment or re-engineering of the feature sets for each new machine learning problem. Therefore, the research agenda for this thesis consists of four main objectives:

I) To analyse whether pre-trained image classification *Convolutional Neural Networks* (CNNs) are suitable for extracting meaningful and robust deep representations from audio spectrograms.

II) To investigate whether deep feature quantisation can effectively reduce noisy representations in the feature space.

III) To verify whether *Deep Convolutional Generative Adversarial Networks* (DCGANs) and recurrent autoencoders are able to learn meaningful deep representations from audio data in an unsupervised manner.

IV) To test whether it is possible to learn both shift-invariant and long-term contextual features from audio signals with *Convolutional Recurrent Neural Networks* (CRNNs).

1.4 Contribution

In contrast to conventional expert-designed feature sets (known as hand-crafted features), and with regard to the rapidly increasing amount of heterogeneous data, this thesis proposes novel deep learning approaches for representation learning directly from audio, hence eliminating the need for hand-crafted features. With respect to the scientific objectives raised above, the main contributions of this thesis are the following:

- Introducing the novel DEEP SPECTRUM system[1] [16], which is an open-source Python toolkit with process parallelisation for rapid GPU-based deep feature ex-

[1] `https://github.com/DeepSpectrum/DeepSpectrum`

traction from audio data utilising pre-trained CNNs. In this thesis, it is demonstrated that the DEEP SPECTRUM features are highly effective for various audio classification and recognition tasks.

- Demonstrating the efficacy of the *Bag-of-Deep-Features* (BODF) proposed for quantising DEEP SPECTRUM representations obtained from real-world audio data. It is shown that the quantisation step, which can be considered as a quasi-filtering process, effectively compresses the feature space and eliminates noisy representations, whilst providing better results in comparison with non-quantised DEEP SPECTRUM features.

- Introducing AUDEEP[2], a novel and highly effective deep architecture for fully unsupervised representation learning from audio data with varying length using recurrent *Sequence to Sequence Autoencoders* (S2SAEs) [17, 18]. This approach has been developed, since commonly used deep representation learning methods such as CNNs, *Restricted Boltzmann Machines* (RBMs), or stacked autoencoders generally require inputs of fixed dimensionality, and do not explicitly model the sequential nature of acoustic data [10].

The deep learning systems introduced in this thesis are validated through extensive experiments on a wide range of audio data, including acoustic sounds, rare audio events, human speech and vocalisation, and medical datasets. Furthermore, DEEP SPECTRUM and AUDEEP have been used as baseline systems for the 2018 edition of the *Audio/Visual Emotion Challenge and Workshop* (AVEC) and the INTERSPEECH *Computational Paralinguistics ChallengE* (COMPARE).

1.5 Outline

This thesis is structured as follows:

Chapter 2 discusses the fundamental principles of *Deep Neural Networks* (DNNs) and the key properties of CNNs and *Recurrent Neural Networks* (RNNs), which form the basis of the higher level models described in the following chapters.

Chapter 3 analyses state-of-the-art representation learning approaches based on pre-trained CNNs, *Generative Adversarial Networks* (GANs), autoencoders, and CRNNs.

Chapter 4 introduces the DEEP SPECTRUM system for rapid extraction of deep representations from audio data utilising pre-trained CNNs.

[2] `https://github.com/auDeep/auDeep`

Chapter 5 proposes a DCGAN structure, which is able to learn meaningful representations from audio data in an unsupervised manner.

Chapter 6 introduces AUDEEP, a S2SAE, which is developed for unsupervised learning of fixed-length representations from variable-length audio data.

Chapter 7 describes a CRNN approach for capturing shift-invariant, high-level features and the long-term temporal context of audio data.

Chapter 8 evaluates the efficacy of the introduced deep representation learning techniques for problems, such as acoustic scene classification, rare acoustic event detection, and audio-based game genre classification.

Chapter 9 analyses the suitability and the performance of the BODF and DEEP SPECTRUM representations for classification of in-the-wild human vocalisation and speech types, and sentiments.

Chapter 10 investigates the applicability of S2SAEs and DEEP SPECTRUM for classification tasks within the health domain.

Chapter 11 concludes the thesis, discusses the limitations of the deep learning methodologies introduced in previous chapters, and suggests avenues for future work.

Part II

Background

2

Deep Neural Networks

Before describing in detail the deep learning methodologies applied in this thesis for the task of representation learning, this chapter will outline the neural network architectures and models that form their basis. In particular, significant miscellaneous deep learning criteria (cf. Section 2.1) and the key properties of two types of neural networks, CNNs (cf. Section 2.2) and RNNs (cf. Section 2.3) will be discussed. Furthermore, four higher level neural network models, which have been applied in the publications leading to this thesis, are introduced. These models are based on pre-trained CNNs (cf. Section 3.1), GANs (cf. Section 3.2), autoencoders (cf. Section 3.3), and CRNNs (cf. Section 3.4).

2.1 Fundamental Principles

An *Artificial Neural Network* (ANN) is a set of algorithms inspired by their biological neural network counterpart, the mammalian nervous system [19]. An ANN is composed of a complex network of interconnected elements, called *neurons*, an information highway that connects multiple input signals into a single output signal. A simple artificial neuron is the McCulloch-Pitts neuron [19] that computes the weighted sum of its bias input and the input signals and passes the result through a non-linear *activation function* (cf. Figure 2.1). For a given McCulloch-Pitts neuron, let there be n inputs and bias b with signals $x_0, \ldots, x_n$ and the weights for each input $w_0, \ldots, w_n$. The input signal x_0 is assigned the bias value b, which sets the weight of the input signal $x_0 = +1$ to $w_0 = b$. The output of the neuron is then:

$$y = f(\sum_{i=0}^{n} w_i x_i) \tag{2.1}$$

where f is the applied activation function.

The information flow between the neurons is defined by the interconnections in an ANN and the weight assigned to each neuron stores the information that the network has acquired. A simple ANN is the *Feedforward Neural Network* (FFNN), in which the

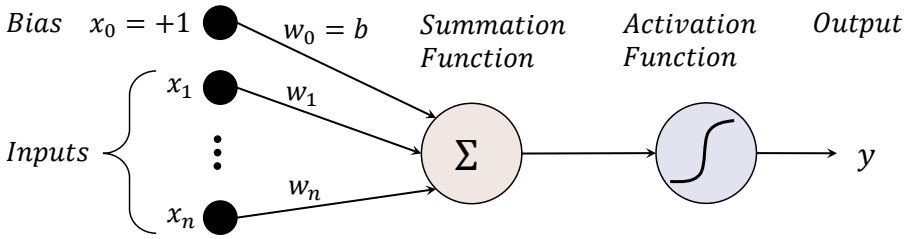


Figure 2.1: A single McCulloch-Pitts neuron with bias b and weights $w_1, \ldots, w_n$ for input signals $x_1, \ldots, x_n$.

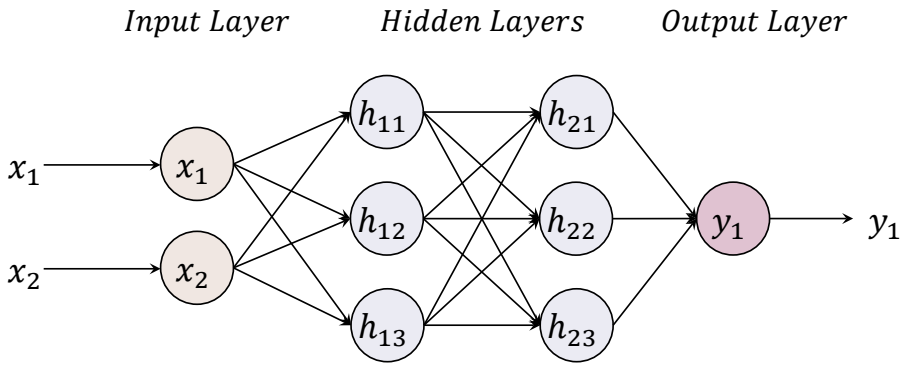


Figure 2.2: An MLP-FFNN with four layers, including one input layer with two neurons, two hidden layers each with three neurons, and one output layer with one neuron. Bias inputs are not depicted for clarity.

information flow is only possible in one direction (i. e. no cyclic connections are allowed): first through the input nodes, then across the hidden units (if some available), and finally through the outputs. An example of such a neural network is the *Multilayer Perceptron* (MLP), which is typically organised into input, hidden, and output layers, each composed of multiple perceptrons. The structure of an MLP with 2 hidden layers, each with three hidden units is given in Figure 2.2.

An ANN composed of two or more hidden layers can be considered as a DNN. Typically, in DNNs, signals pass through multiple hidden layers and flow across a large number of neurons before reaching the output layer. The structure of a DNN is given in Figure 2.3. In the following sections, important characteristics of DNNs are briefly described.

2.1.1 Network Training

Once a DNN has been structured for a specific application, it is ready for the training process, which can be broadly categorised into *supervised* and *unsupervised* approaches.

- *Supervised Training:* The network has access to both input and output, i. e. the data

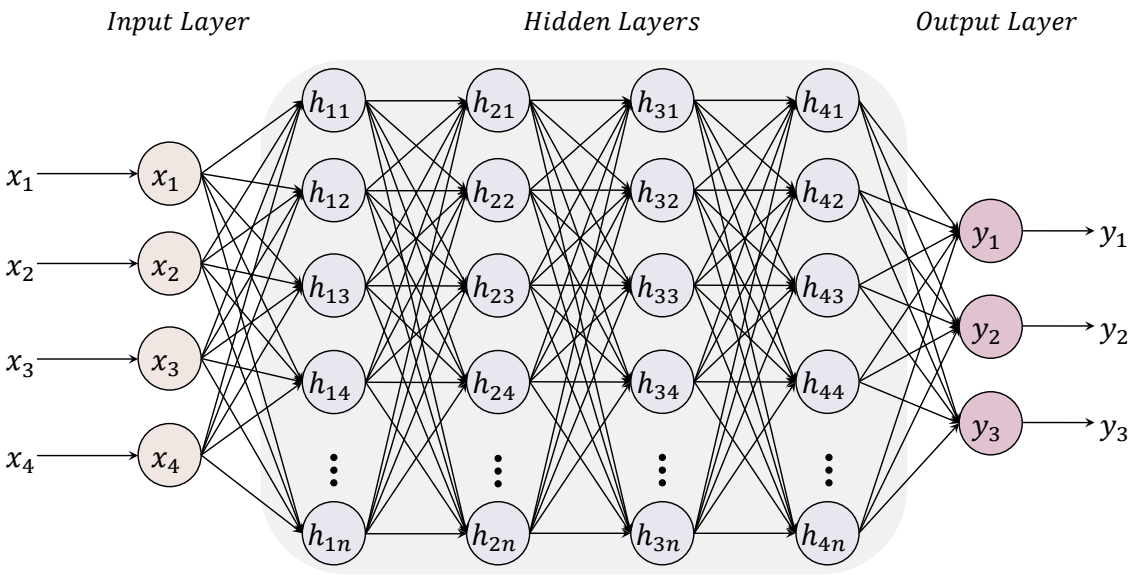


Figure 2.3: A high-level structure of a DNN with three hidden layers.

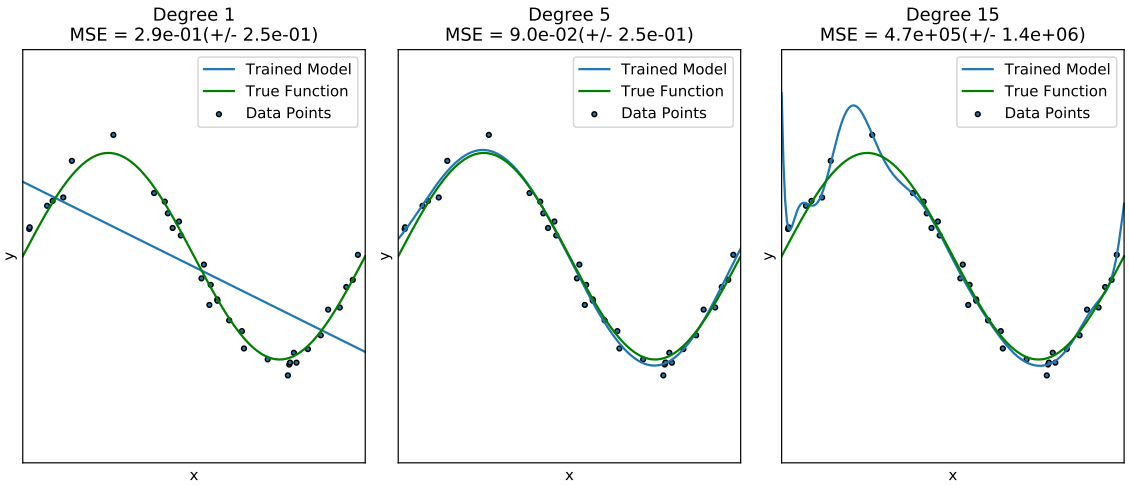
is fully labelled. Hence, it is possible for the network to update the weights leading to the neurons using *backpropagation*, i. e. by propagation of the error backwards from the output layer to the hidden layer [19–23]. A *loss function* computes the error given at the output layer using the difference between the network output and the expected output at each output unit.

- *Unsupervised Training:* The network is provided with the inputs, but it has no access to the desired target outputs, i. e. there is no label information available. In this case, the neural network discovers regularities, patterns, or categories in the input data and learns a distribution based on which a loss function determines the likelihood between input and output signals.

2.1.2 Network Hyperparameters

In a DNN, hyperparameters determine the network structure and define the way in which it is trained. Unlike the weights and biases of a DNN, hyperparameters are not model parameters and cannot be directly trained from the input data. In the following, some of the important hyperparameters related to network structure and training algorithm are briefly introduced:

- *Learning rate:* Defines the rate at which the learning process reaches the local minimum. The lower the learning rate, the slower is the training process, however, the network converges smoothly. High learning rate speeds up the training at the cost of convergence.



(a) Underfitting (b) Good fit (c) Overfitting

Figure 2.4: A graphical representation of the underfitting and overfitting problems. Figure 2.4a shows a linear function, which is not sufficient to fit the data points. Figure 2.4b illustrates a polynomial of degree 5, which approximates the true function very accurate. In Figure 2.4c, the trained model is overfitted to the data points for higher polynomial degrees. The diagrams are generated using implementations provided in the scikit-learn machine learning library [24]. The overfitting and underfitting are evaluated quantitatively by using a *Cross Validation* (CV). The MSE on the validation set is calculated; the model generalises better for lower MSE values.

- *Momentum:* Is an addition to classic *Stochastic Gradient Descent* (SGD) algorithm, which helps accelerating gradients vectors in the right directions, hence leading to faster convergence [25–27]. It also assists in the prevention of oscillation at the local minimum [25–27]. Many state-of-the-art deep learning models use SGD with momentum for training [28–30].

- *Batch size:* Is the number of training samples utilised in one training iteration. Very small batch size can lead to stochastic behaviour of the network and has a similar effect as choosing high learning rate. Large batch size can generate more stable results, but choosing a very large batch size can yield worse performance.

- *Number of epochs:* Is the number of iterations that the entire input data is passed forward and backward through the neural network. The higher the number of epochs is, the higher is the number of times the weights are changed in the network. By increasing number of epochs, the trained model goes from *underfitting* (cf. Figure 2.4a) to *a good fit* (cf. Figure 2.4b), and then to *overfitting* (cf. Figure 2.4c). There is no general rule to find the best number of epochs in a training process [19]. Underfitting means that the trained model is too general and can neither model the

input data nor generalise to new data. On the contrary, an overfitted model reflects the errors in the input data on which it is trained, rather than accurately recognising new data.

- *Number of hidden layers and units:* Defines the complexity of the neural network. By adding one or more hidden layers, the network is able to extract higher-level features from its input data [19]. The number of hidden units can be roughly set to equal the number of input features. A small number of hidden units may be an additional cause of underfitting [19].

2.1.3 Regularisation

Regularisation is a set of techniques used to avoid overfitting. It helps the network to train a model from the input data which is more robust and can generalise better to unseen data. In the following, some of the most important regularisation approaches are briefly introduced. For detailed information, interested reader is referred to corresponding references.

- *More data:* The best regulariser is more data. If the training data is scarce, data augmentation techniques can be applied. For audio data augmentation, methods like *Stochastic Feature Mapping* (SFM) [31], which is inspired by the idea of voice conversion [32–34], *Vocal Tract Length Perturbation* (VTLP) [35], time stretching, or pitch shifting of audio signals have shown to be effective. For image data augmentation, state-of-the-art approaches, such as GANs have shown their strength to generate synthetic samples based on the training data [36, 37]. These methods can also be applied on audio spectrograms.

- *Dropout:* Is one of the most popular and effective regularisation techniques to cope with overfitting problems [38, 39]. Dropout reduces interdependent learning amongst the neurons in the network by dropping units with their connections from the neural network during training. In Figure 2.5, an example of a thinned neural network produced after applying dropout has been shown.

- *DropConnect:* Is a generalisation of dropout [38, 39] for regularising large fully connected layers and preventing "co-adaption" of units in a neural network [40]. The main difference between DropConnect [40] and dropout is that individual weights in the network are deactivated instead of disabling neurons (cf. Figure 2.6). Results provided by Wan et al. [40] on a range of datasets demonstrated that DropConnect often outperforms dropout.

- *Batch Normalisation* (BN): Is a technique to normalise the internal representation of the input data to boost the training of the neural networks [41]. Furthermore, BN regularises the training model and reduces the need for dropout [38].

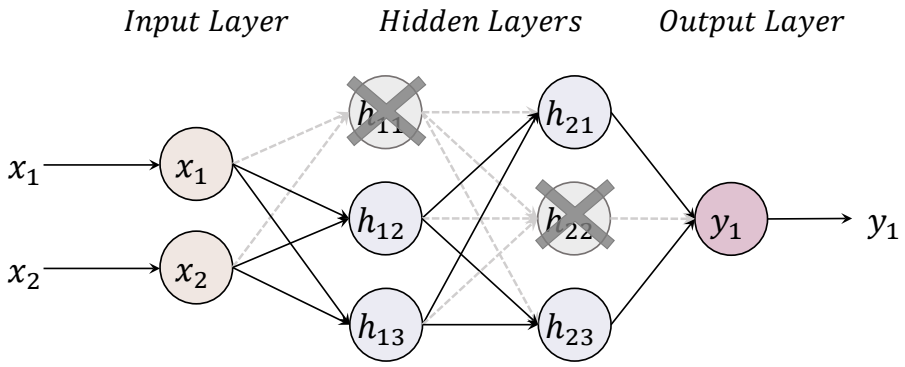


Figure 2.5: The structure of a 'thinned' MLP-FFNN after applying dropout. Crossed neurons and their connections (dashed arrows in grey) have been dropped.

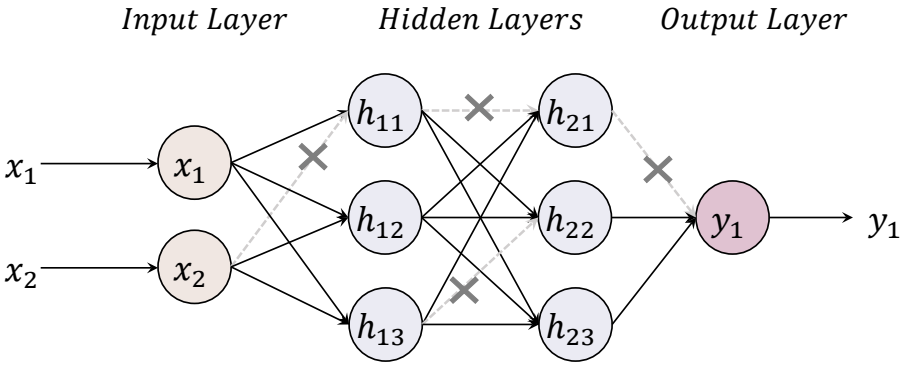


Figure 2.6: The structure of an MLP-FFNN after applying DropConnect. Individual weights are set to zero (crossed dashed arrows), instead of units. In this case, a neuron can remain partially active.

- *Early stopping:* Is a form of regularisation to minimise the potential overfitting problems by early stopping the optimisation process if the performance of the network has not improved for a given amount of epochs.

2.1.4 Activation Functions

Activation functions are mathematical functions that define the output of a neuron referring to its input value. Without an activation function, the output of the network would be a linear function (polynomial of degree one), which is limited in its ability to learn complex structures from an input signal (cf. Figure 2.4a). With linear activation functions, cumulative back-propagated error signals either grow out of bounds, or shrink very fast [23]. Hence, differentiable non-linear activation functions are required to make backpropagation possible and learn robust representations from non-linearly separable data [42]. Some of the most important non-linear activation functions are defined below:

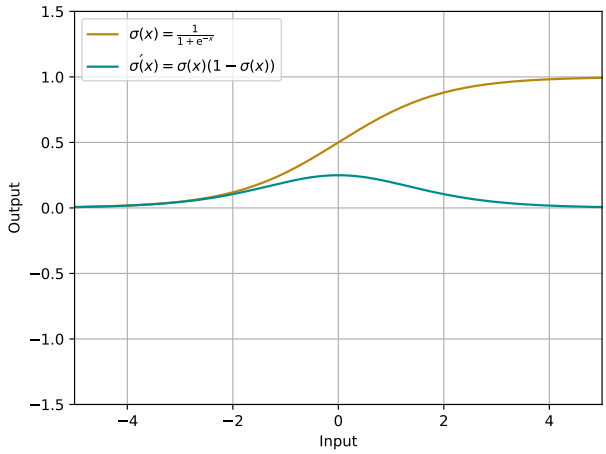


Figure 2.7: Sigmoid activation function and its derivative.

- *Sigmoid*: The *sigmoid* function is a popular activation function that often finds application if a neuron's output should be classified as one of two values. It is defined as follows:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2.2}$$

This function is smooth and continuously differentiable, and its values range from zero to one. Therefore, for the purposes of backpropagation, its derivative can be determined as the following function:

$$\sigma'(x) = \sigma(x)(1-\sigma(x)) \tag{2.3}$$

Figure 2.7 shows the *sigmoid* function together with its derivative.

- *Tanh*: Certain characteristics of the sigmoid function can be seen as limiting, for example, its range is restricted to positive values between zero and one. The *tanh* activation addresses this limitation by scaling and shifting the *sigmoid* function, so that the values are in the range of $[-1,1]$:

$$tanh(x) = \frac{2}{1+e^{-2x}} - 1 \tag{2.4}$$

Its derivative is still easily computed as:

$$tanh'(x) = 1 - tanh(x)^2 \tag{2.5}$$

Figure 2.8 shows the *tanh* function together with its derivative. The *tanh* activation function is prominently applied in RNNs using *Long Short-Term Memory* (LSTM) cells [43] or *Gated Recurrent Units* (GRUs) [44].

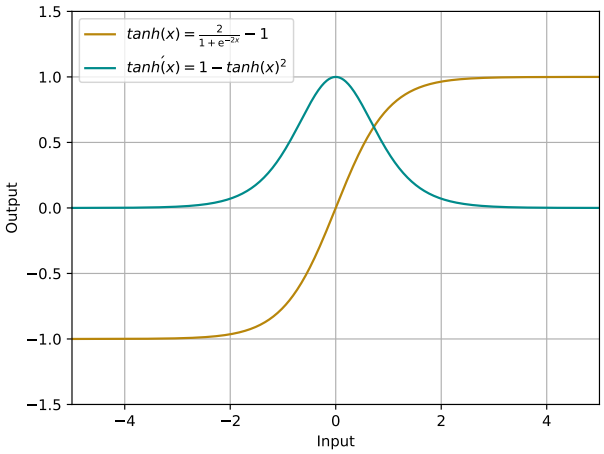


Figure 2.8: Tanh activation function and its derivative.

- *Rectified Linear Unit* (ReLU): A ReLU [45] is a very simplistic, yet surprisingly efficient activation function for introducing non-linearity to a neural network architecture. A ReLU is defined by the identity function for values equal to and above zero and squashes all negative values to zero:

$$R(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \tag{2.6}$$

  This also means that not all neurons on a layer necessarily get activated all the time, making computation both sparser and faster. The derivative of the ReLU is zero for $x < 0$ and one for $x \geq 0$:

$$R'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases} \tag{2.7}$$

  ReLUs are one of the most popular activation functions used within modern neural network architectures. They have a wide range of application within state-of-the-art machine learning models. Networks trained for image recognition tasks, such as AlexNet [28] or ResNet [46] use ReLUs as their activation functions. For the deep learning models introduced in this thesis, ReLUs have been applied as activation function in the hidden layers.

  Figure 2.9 shows the ReLU function together with its derivative.

- *Leaky* ReLU: Leaky ReLUs [47] modify regular ReLUs by allowing a small non-zero gradient for units with negative activations. This is done by multiplying activations below zero with a fixed, small factor α:

$$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \tag{2.8}$$

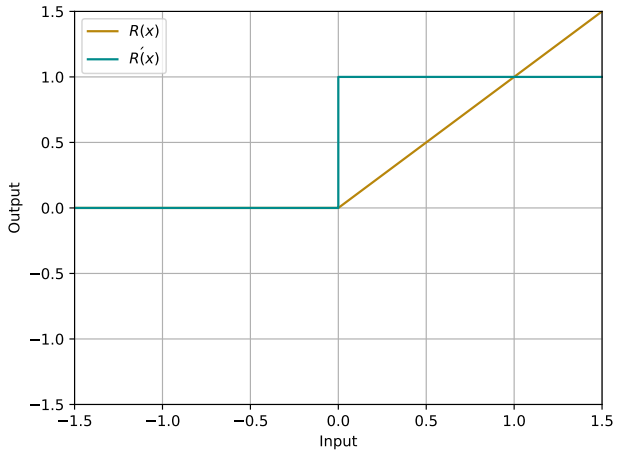


Figure 2.9: ReLU activation function and its derivative.

The derivative therefore becomes:

$$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases} \tag{2.9}$$

Leaky ReLUs activations have been applied in the discriminator element of DCGANs [48]. Figure 2.10 shows the *leaky* ReLU function together with its derivative.

- *Softmax*: In contrast to the rest of the discussed activation functions, the *softmax* function was specifically designed for multi-class classification problems and is therefore used in the last layer of a neural network architecture. It transforms the activations of the neurons on a single layer to class probabilities values between zero and one, which sum up to one. The softmax activation of the j-th of K neurons on the same layer is defined as follows:

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}} \quad \text{for } j = 1, \ldots, K. \tag{2.10}$$

2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are the first neural network architecture which is considered for representation learning in this thesis. CNNs gained in popularity in the machine learning community for their efficacy in solving visual recognition tasks, such as image and video classification [28, 30, 46, 49] and action or face recognition [50–54]. Classic CNNs make use of two distinct types of hidden layers: the eponymous *convolutional* layers and *pooling* layers.

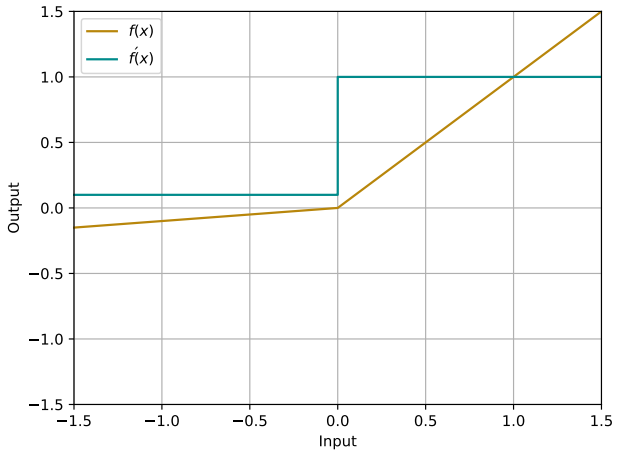


Figure 2.10: Leaky ReLU activation function and its derivative.

Convolutional layers convolve their inputs with striding kernels of a specific small size, resulting in feature maps. The parameters of each kernel are shared between convolutions with different parts of the input. As a result, the network is able to recognise different features of the input, such as detecting edges, regardless of where they are located in the original image.

Conversely, pooling layers reduce the number of neurons in feature maps by putting together groups of adjacent neurons. Furthermore, they are commonly deployed to shrink the input image size, in order to achieve shift-invariance [55]. Thus, the computational load, the memory usage, and the number of trainable parameters can be reduced [56, 57]. In addition to limiting the risk of overfitting, a smaller input image size also improves the tolerance towards *location invariance* [55, 57]. One of the most popular pooling approaches is *max-pooling*, through which small square regions of the feature maps are reduced by the maximum activation in this region [56]. Conventional fully connected layers are also often added after these two special layers, and, in the case of classification tasks, a softmax layer can be used to complete the architecture. Figure 2.11 shows the structure of a multilayer CNN.

Apart from the field of computer vision, CNNs have also been successfully applied in other research areas. In *Natural Language Processing* (NLP), they have been used for tasks, such as sentence classification [58, 59] or sentiment analysis [60–62], whilst they also achieve state-of-the-art results in computer audition when applied to spectrograms, for example in environmental sound [63, 64], or acoustic scene classification [65, 66].

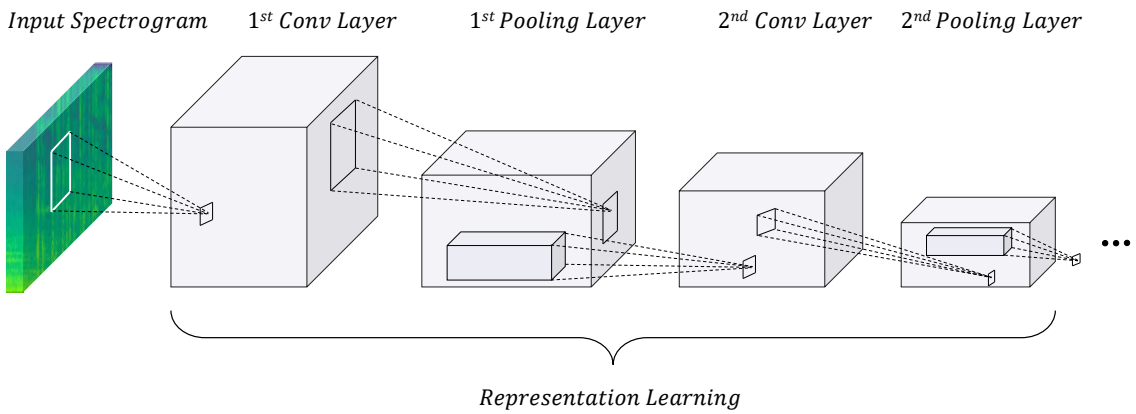


Figure 2.11: Architecture of a multilayer CNN. Kernels are convolved with the input to create feature maps in the convolutional layers. Pooling is used as a dimensionality reduction technique afterwards.

2.3 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a type of ANN that aim at modelling sequential data, such as continuous speech and text data [67, 68]. The structure of a standard RNN is similar to a simple MLP model, yet introduces connections between hidden layers of various time steps[1].

These connections can be of three different forms. If the neurons are connected to themselves between each time step, the connection type is *direct*. Neurons can also be connected to neurons on the same (*lateral*) or different (*indirect*) hidden layer. This enables the network to "learn from the past", i. e. discovering temporal correlations and dependencies in the data distribution. Training of RNNs can be achieved by *Backpropagation Through Time* (BPTT), which is conceptually similar to 'unrolling' the recurrent connections into a multilayer DNN consisting of copies of the network for each time step.

Regular RNNs are comparably difficult to train, as the BPTT algorithm suffers from *vanishing* and *exploding* gradients, which hinder the separation of correlating similarities between data points by longer periods of time [69–71]. In order to tackle this problem, an RNN architecture with *Long Short-Term Memory* (LSTM) cells is designed [43, 72]. LSTMs enforce constant error flow through the use of memory cells in combination with gate units. A memory cell stores a hidden state, whilst multiplicative gate units control how this state is influenced by the current input (*input gate*), as well as how it should affect the current output (*output gate*). Figure 2.13 demonstrates this architectural concept extended with *forget gates*, which were introduced to help LSTMs handle very long or continuous sequences without pre-defined beginning and end. These gates facilitate the network resetting its internal state at a certain time.

[1] A time step is an easy way to distinguish between the internal states of a neuron in an RNN.

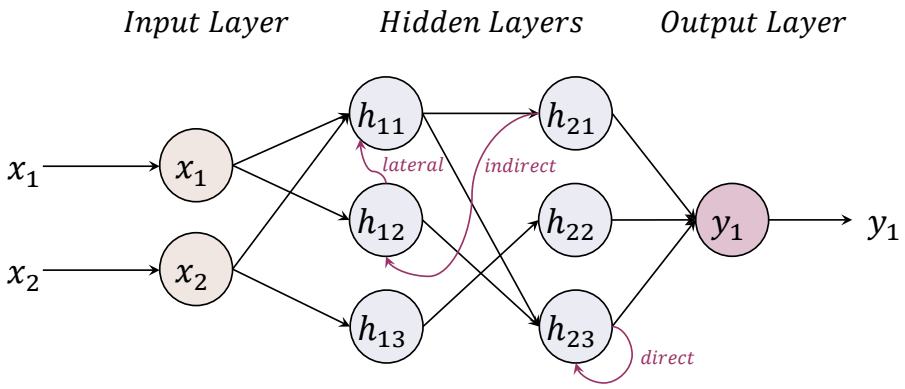


Figure 2.12: An RNN architecture with three different types of recurrent connections (red arrows). *Direct* connections link a neuron to itself between time steps. Connections between different neurons are called *lateral* connections when the neurons reside on the same hidden layer, and *indirect* connections if they are on separate layers.

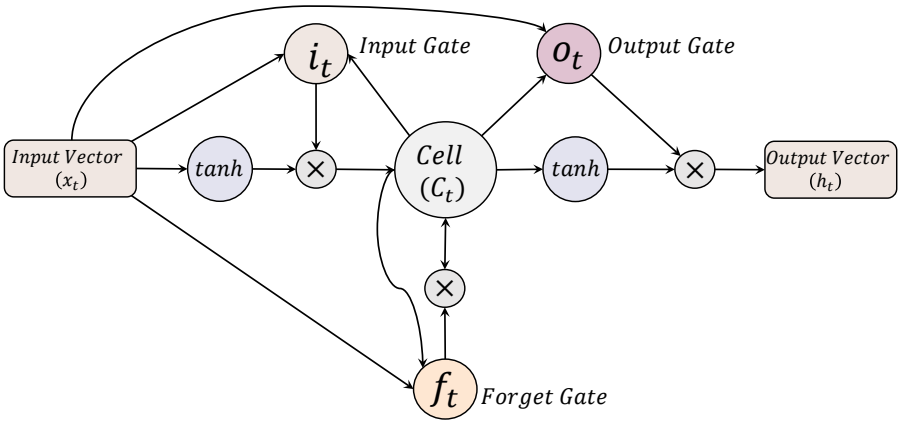


Figure 2.13: Basic building block of the LSTM architecture. A memory *cell* stores a hidden state, whereas different *gated units* regulate how this state should be affected by the input (*input gate*), influence the output (*output gate*), or even be forgotten after certain events (*forget gate*).

3

Representation Learning

In this chapter, four higher level neural network models based on CNNs and RNNs are introduced and their application areas are discussed. These models form the basis of the deep representation learning methodologies applied in this thesis. In particular, pre-trained CNNs (cf. Section 3.1), GANs (cf. Section 3.2), autoencoders (cf. Section 3.3), and CRNNs (cf. Section 3.4) are investigated.

3.1 Pre-trained Convolutional Neural Networks

As described in Section 2.2, many state-of-the-art CNN architectures are deep, featuring large amounts of trainable parameters. Whilst this allows them to perform very well on sizeable amount of training data, e. g. for CNNs trained on the ImageNet corpus [73] or Google's [74] and Facebook's [75] face recognition networks, they often fall behind models using hand-crafted features [76] when applied to smaller datasets. A possible solution to utilise the advantages of these deep networks when training data is sparse, is transferring the knowledge learnt from large datasets [77, 78]. In such cases, the activations of the neurons extracted from the fully connected layers of pre-trained image CNNs can be directly applied as off-the-shelf features for a diverse range of visual recognition challenges, often outperforming hand-crafted feature sets [79].

Chen et al. [80] use feature vectors extracted from a CNN pre-trained on the CASIA-WebFace dataset for unconstrained face recognition [81]. Marmanis et al. perform classification on observations of the earth by feeding features extracted from an ImageNet pre-trained CNN into their own, smaller CNN classifier [82]. Venugopalan et al. propose a deep end-to-end sequence-to-sequence model, which uses the outputs of both an action and object pre-trained CNN as inputs for a stacked LSTM to translate videos to text [83].

Another approach that can often lead to even better results when using pre-trained CNNs is to fine-tune their weights on the target dataset. This transfer learning method is widely applied to various tasks, from visual emotion recognition [84] over cross-modal retrieval [85] to medical applications, such as computer aided detection, or mammogram

analysis. Kieffer et al. [86] further compare the efficiency of training a CNN network from scratch and fine-tuning a pre-trained CNN for the task of histopathology classification. Pre-trained CNNs are also used to initialise components of the fast and faster R-CNN models used for object detection [29, 87].

Apart from visual recognition, pre-trained CNNs have also recently found applications in computer audition. The author and his colleagues utilised image CNN descriptors extracted from spectrogram plots for audio-based recognition tasks [16, 62, 88]. Aytar et al. train *SoundNet* – a deep CNN architecture providing sound representations – on unlabelled video data collected in-the-wild by utilising the recognition power of established visual CNNs [89]. The representations learnt by SoundNet have since been applied to various tasks. Grinstein et al. made a foray into audio style transfer [90], whereas IBM uses the features as part of their sports highlights system to detect excitation in the commentator's voice [91]. Hori et al. use SoundNet features in their multimodal attention system that fuses audio and spatio-temporal features to generate video descriptions [92]. Researchers at Google have further investigated the effectiveness of traditional image recognition CNN architectures for audio analysis. They train model variations on a large dataset of 70 million automatically labelled YouTube videos and later evaluate the embeddings learnt by these models on the *AudioSet* [93] ontology for acoustic event classification [94].

3.2 Generative Adversarial Networks

The second representation learning technique investigated in this thesis utilised *Generative Adversarial Networks* (GANs). In 2014, Goodfellow et al. first introduced the GAN framework [95]. In GANs, two different networks are trained simultaneously in a zero-sum game-like manner. A *Generator* creates samples from a random distribution (typically a noise vector z), whereas a *Discriminator* is trained to distinguish these generated samples from real data. This adversarial setting leads to the need of both models to continually increase their performance, until the samples produced by the generator become indistinguishable from the real data distribution [95], or at least an equilibrium is reached, in which both models cannot improve further.

In the framework proposed by Goodfellow et al. [95], a prior probability distribution p_z is characterised over the input noise variables z, which is then mapped into the data space by a differentiable generator function $G(z)$. In [95], $G(z)$ is represented by an MLP. In addition, a second MLP, with a differentiable discriminator function $D(x)$, is defined. $D(x)$ calculates the probability that a given sample x was drawn from the data distribution p_{data}, instead of the generator distribution p_g. Afterwards, the discriminator D is trained with backpropagation to maximise the probability of assigning the correct label to both training examples and generated samples from G. At the same time, the generator G is trained to minimise $\log(1 - D(G(z)))$, i. e. to maximise the number of generated samples which are misclassified by the discriminator. It has been demonstrated

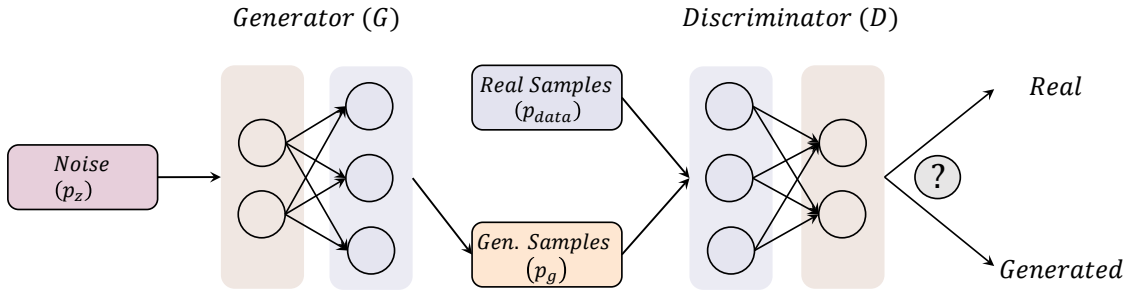


Figure 3.1: Basic concept of a GAN. The *generator G* constructs samples from noise z according to the learnt distribution p_g. The *discriminator D* is then tasked with differentiating whether a sample was created by the generator or drawn from the real data p_{data}.

that the minimax game has a global optimum at $p_g = p_{data}$ when the generator has learnt to perfectly reproduce the real data distribution [96]. After the training process is finished, the activations in the discriminator can be extracted as representations for the real training examples [48]. Figure 3.1 illustrates the basic concept of the introduced GAN.

Chen et al. extend GANs from an information-theoretical point of view with *Information Maximising Generative Adversarial Networks* (InfoGANs) [97]. InfoGANs learn interpretable representations in an unsupervised manner by maximising mutual information between a set of latent variables and the generator distribution. The noise vector is split up into a source of incomprehensible noise z and a set of latent variables, denoted as latent code c. This code targets structured salient information of the data distribution, e. g. for the MNIST database of handwritten digits [98], individual variables of c learn to represent the digit kind, the width, or the rotation of the generated character.

Wasserstein-GANs [99] minimise an efficient approximation of the Earth Mover distance between generated and real data distribution in order to effectively combat a problem found with training of GANs. The problem they have addressed is that GANs require balanced training of generator and discriminator and are quite sensitive to changes in the network architecture [100].

Whilst the samples generated by a GAN are often indistinguishable from the real data distribution by the human eye, Valle et al. showed that these fake samples carry a unique signature that makes them easily identifiable using methods of statistical analysis and pixel value comparison [101]. The samples also violate formal specifications that can be learnt from the respective real data.

3.3 Autoencoders

Autoencoders are neural networks that are able to learn compressed and efficient data coding by unsupervised training. The encoder part of the network maps the input data to a

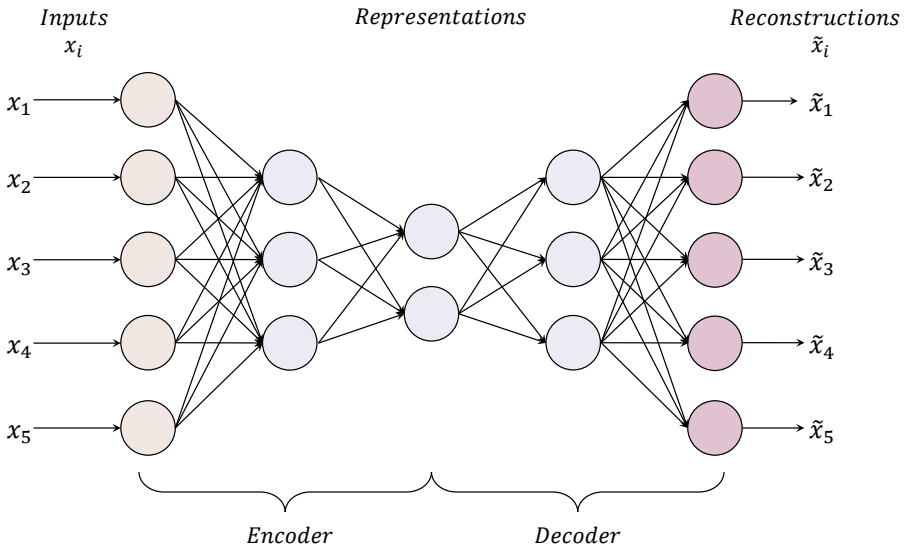


Figure 3.2: Illustration of a feedforward autoencoder. The encoder maps the inputs $x_1, \ldots, x_n$ to hidden representations. Afterwards, the decoder reconstructs $\tilde{x_1}, \ldots, \tilde{x_n}$ from this representation. The accuracy of the representation is evaluated using an objective function, such as the MSE.

hidden representation, normally of lower dimensionality, e. g. by stacking fully connected layers that decrease in neuron count. The decoder part of the network then attempts to reconstruct the input data from this compressed representation. The training goal of the network is therefore to minimise the difference between the original data it receives as input and the output it then reconstructs from the learnt coding. For this purpose, the *Mean Squared Error* (MSE) is often used between the network inputs $x_1, \ldots, x_n$ and the reconstruction $\tilde{x_1}, \ldots, \tilde{x_n}$

$$e_{MSE} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \tilde{x_i})^2$$

as the objective function during the training process [102]. A simple form of an autoencoder is an MLP architecture having an input and output layer of the same size and hidden layers in between (cf. Figure 3.2).

There is a range of autoencoders with the aim to learn richer representations and prevent the networks from learning the identity function. Noteworthy are denoising, or stacked denoising autoencoders [102, 103], in which the input data is intentionally corrupted before being fed into the encoder network. The autoencoder is then trained to reconstruct the original, clean data from the corrupted samples [102].

3.4 Convolutional Recurrent Neural Networks

In a CRNN, local dependencies of the input signals are extracted by adopting a CNN, whilst the global structures are obtained with recurrent layers. Such a system was first proposed in [104] for text classification and then applied for image recognition [105–107], and audio and music classification tasks [108–110]. CNN layers are able to learn high-level, shift-invariant features, and help recurrent layers to learn robust spatial dependencies from the visual inputs (e. g. from images for object recognition or spectrograms for speech processing) [107, 108]. A CRNN system has less complexity compared to an RNN-only approach, as the integrated CNN provides abstraction and reduces the overall number of trainable parameters. Moreover, the long-term temporal context encoded by recurrent layers can transfer more precise supervisions to the CNN layers during back-propagation.

In the context of visual recognition, Wu et al. use a CNN in combination with a recurrent layer that operates as a conditional random field for real-time road object segmentation [111]. Furthermore, Wehrmann et al. detect adult content in videos with CRNNs [112].

Xu et al. achieved strong results for audio event detection using gated CNNs [113]. For the similar task, the authors in [114–117] successfully applied end-to-end CRNN approaches with state-of-the-art results. For the challenge of bird audio detection, CRNNs have also been successfully applied [118, 119]. The CRNN approach proposed by Iqbal et al. [120] extended the VGG13 [30] architecture, in which they initially averaged only the frequency dimension instead of averaging across the spatial dimensions after the convolutions. Afterwards, they applied a bidirectional recurrent layer for each time step, in order to learn the temporal dynamics of the input signals. Iqbal et al. achieved robust results with their system for the task of *general-purpose audio tagging* [120].

CRNNs have also been used for a number of health care and wellbeing applications. Gao et al. use them to grade nuclear cataracts (clouding of the lenses in the human eye) [121], whereas Qin et al. reconstruct magnetic resonance images [122]. Furthermore, Ma et al. build a deep CRNN architecture that learns to effectively represent patients' electronic health records, in order to predict health risks more precisely [123]. Multimodal activity recognition from data collected with wearable devices has also been performed with CRNNs by Ordonez et al. [124].

In genomics, Qang and Xie have applied a convolutional bidirectional LSTM network to DNA sequences for quantifying their function [125]. Pan et al. predict RNA-protein sequences and structure binding preferences using a CRNN architecture [126].

The author of this thesis and his colleagues have successfully applied a CRNN with *Bidirectional Long Short-Term Memory* (BLSTM) cells in the recurrent layers, for audio-based recognition of echolalic vocalisation from children with an *Autistic Spectrum Condition* (ASC) [127], and a CRNN with LSTM cells (with backward pass) in the recurrent layers, for rare acoustic event detection [128].

Part III

Methodologies

4

Deep Spectrum

To solve complex, real-world recognition tasks, machine learning systems should be given a considerable amount of prior knowledge to compensate for data which is not available. CNNs are able to combat this with currently established models [28,129–132]. Therefore, as described in Section 3.1, it should be possible to transfer the knowledge of deep CNNs that have been pre-trained on large-scale image datasets (e. g. ImageNet [73]) to computer audition and vision related tasks, in which the data is scarce, such as medical image analysis [133–135], or text classification [59]. In order to make use of the robustness of pre-trained CNNs for audio processing tasks, the author and his colleagues have developed the DEEP SPECTRUM system[1] [16], which is an open-source Python toolkit with process parallelisation for rapid GPU-based deep feature extraction from audio spectrograms.

4.1 Characteristics

The DEEP SPECTRUM system has been successfully applied for medical audio processing [16, 88, 136, 137], affective computing [4, 62], and human speech and vocalisation tasks [88, 138]. It has also been used as a baseline system for the 2018 edition of the *Audio/Visual Emotion Challenge and Workshop* (AVEC) [139]. A step-by-step tutorial on how to perform feature extraction with DEEP SPECTRUM, and a complete documentation of the DEEP SPECTRUM command line interface is provided on the GitHub repository page.

4.2 Architecture

An overview of the DEEP SPECTRUM system is given in Figure 4.1. In the pre-processing step, two-dimensional visual representations of the input audio files (e. g. spectrograms or

[1] `https://github.com/DeepSpectrum/DeepSpectrum`

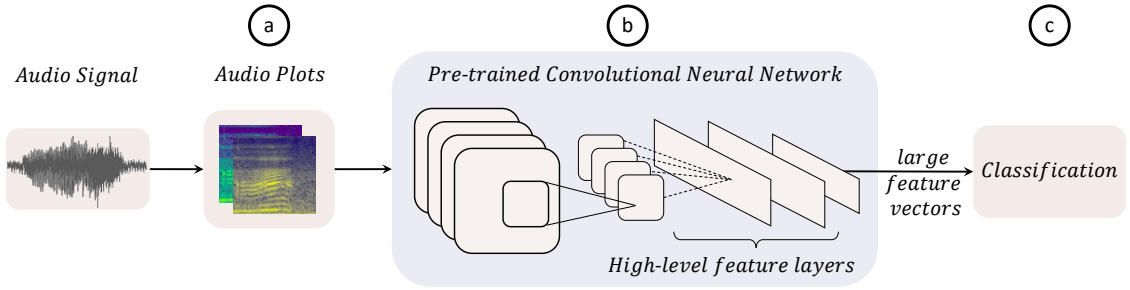


Figure 4.1: Illustration of the DEEP SPECTRUM system. Pre-trained CNNs are used to obtain task-dependent representations from the audio spectrograms. A detailed description of the system architecture is given in Section 4.2.

mel-spectrograms) are created (cf. Figure 4.1a). This step is necessary as the CNN descriptors use two-dimensional filters to process the input spectral representations (cf. Figure 4.1b). After forwarding the spectrograms through pre-trained CNNs, the activations of the fully connected layers of each network are then extracted as feature vectors. These high-level, shift-invariant features, denoted as DEEP SPECTRUM features, are used to train a classifier (cf. Figure 4.1c). It is worth mentioning that the convolutional layers are able to make strong assumptions about the locality of pixel dependencies, i. e. the more local structures are available in the generated visual representations, the more robust are the extracted DEEP SPECTRUM features.

4.2.1 Creation of Audio Plots

As the pre-trained CNNs accept two-dimensional images as input, the first step is to create audio plots from the input audio signals. Using DEEP SPECTRUM it is possible to create spectrograms, mel-spectrograms, or chromagrams, and their derivatives, which are then sent through the CNNs to extract the DEEP SPECTRUM features. Spectrograms are computed from Hanning windows of width w and overlap of αw, where $0 < \alpha < 1$ describes the percentage of the overlapping window. The Hanning window helps to preserve both the frequency resolution and the amplitude of a signal [140, 141].

Mel-spectrograms are calculated from the log-magnitude spectrum by dimensionality reduction using a mel-filter with N_{mel} filterbanks equally distributed on the mel-scale defined in eq. (4.1):

$$f_{mel} = 2595 \cdot \log_{10}\left(1 + \frac{f_{Hz}}{700}\right), \tag{4.1}$$

where f_{mel} is the resulting frequency on the mel-scale computed in mels and f_{Hz} is the normal frequency measured in Hz. The mel-scale is based on the frequency response of the human ear that has better resolution at lower frequencies (as compared to a linear representation). The mel-spectrogram is also displayed on this scale.

Chromagrams are a mapping of the spectrogram bins into structures based around pitch relationships, as defined by the western music tonality system. They also characterise the pitch class content of audio signals over time [142]. Compared to spectrograms, chroma features relate to the 12 pitch classes defined in the twelve-tone equal temperament which are represented by attributes of pitch: *C, C#, D, D#, E, F, F#, G, G# A, A#,* and *B*. Furthermore, the first order derivatives (deltas) of the mel-spectrograms and chromagrams to incorporate more of the dynamics of the underlying features can be computed.

For creation of all audio plots, implementations provided by the *librosa*[2] Python library are used. The plots are also scaled and cropped to square images without axes and margins to comply with the input needed by the pre-trained CNNs. The audio plots have an intermediate size of 387×387 pixels, and are then further scaled down to 227×227 pixels for AlexNet (cf. Section 4.2.2.1) and 224×224 pixels for VGG networks and GoogLeNet (cf. Sections 4.2.2.2 and 4.2.2.3). Figure 4.2 shows the visual differences between the mentioned audio plots extracted from a five second speech signal.

The author and his colleagues also demonstrated that alternative DEEP SPECTRUM features will be extracted when the colour maps are changed for the same input audio plots [16, 88]. Based on the findings in [4, 16, 62, 88], the DEEP SPECTRUM representations extracted from the audio plots with *viridis* colour map show stronger performance than a group of other colour maps, including *cividis*, *hot*, *magma*, *plasma*, or *vega20b*. The reason for this effect could be the spectrum of colour available with *viridis* – which is a perceptually uniform sequential colour map, changing from blue (low range) to green (mid range) to yellow (upper range) (cf. Figure 4.3) – covering a wide range of colours available from the ImageNet training images.

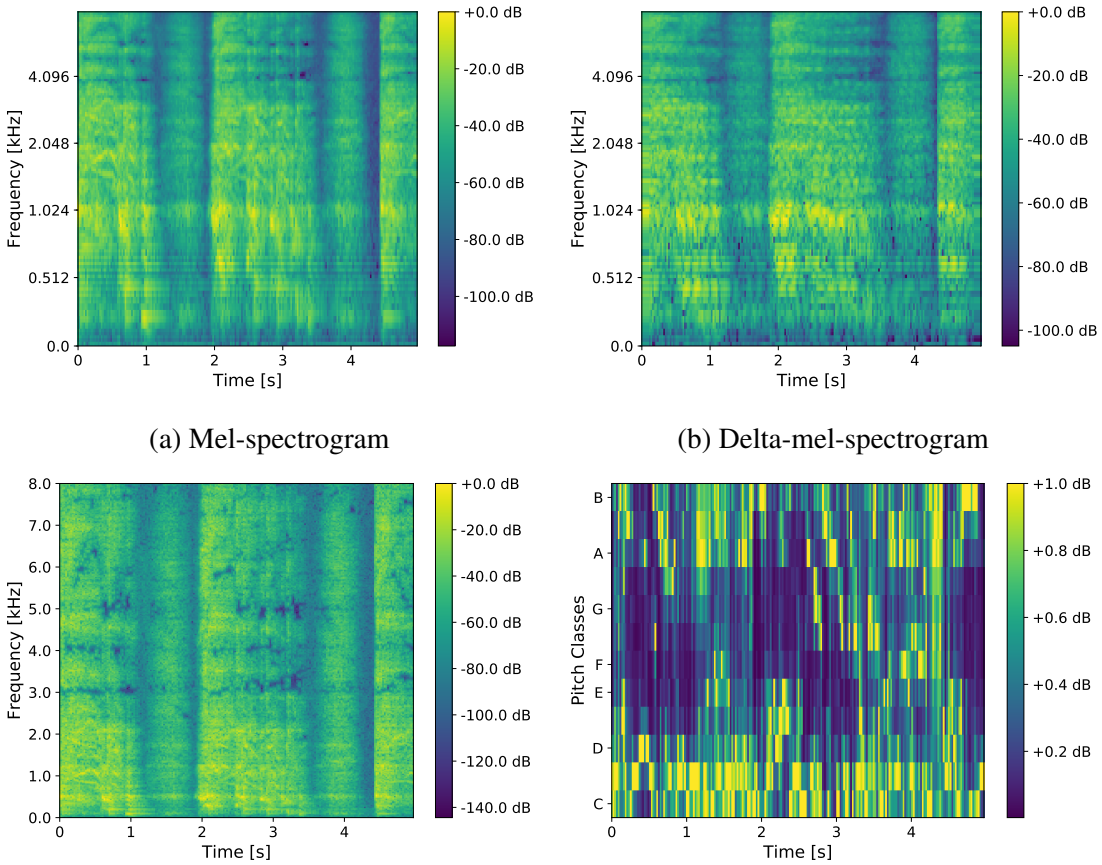
4.2.2 Applied Pre-trained CNNs

To form suitable deep representations from audio plots (cf. Section 4.2.1), four pre-trained CNNs, AlexNet [28], VGG16 and VGG19 [30], and GoogLeNet [144], are integrated in the DEEP SPECTRUM system and can be applied as feature extractors. The architectures of AlexNet and VGG networks are compared in Table 4.1. The structure of the inception module used in GoogLeNet is shown in Figure 4.4.

4.2.2.1 AlexNet

AlexNet has 5 convolutional layers, in cascade with 3 fully connected layers [28]. An overlapping max-pooling operation is applied to downsample the feature maps generated by the first, second, and third convolutional layers. A ReLU non-linearity is used, as this non-saturating function regularises the training, whilst improving the network's general-

[2] `https://librosa.github.io/librosa/`



(a) Mel-spectrogram

(b) Delta-mel-spectrogram

(c) Spectrogram

(d) Chromagram

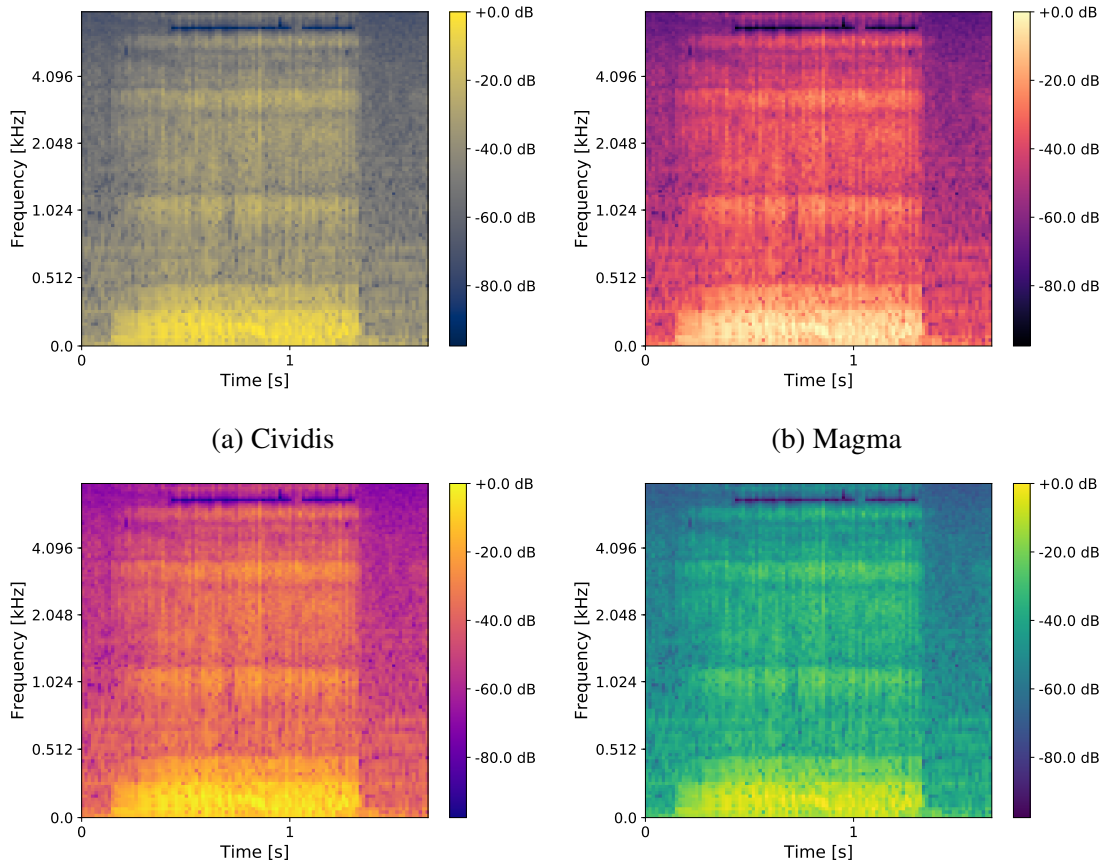
Figure 4.2: Example mel-spectrogram, delta-mel-spectrogram, spectrogram, and chromagram of a 5 second speech signal.

isation capabilities[3]. The fully connected layers *fc6*, *fc7*, and *fc8*, have 4 096, 4 096, and 1 000 neurons, respectively. The DEEP SPECTRUM features are then obtained from the activations of the neurons in the fully connected layers.

4.2.2.2 VGG16/VGG19

Whilst the filter sizes change across the layers in AlexNet, VGG16 and VGG19 have on the contrary a constant 3×3-sized receptive field in all of their convolutional layers [30]. Both deep architectures consist of 2 more max-pooling layers in comparison with AlexNet and have deeper fully connected layers in cascade. Similar to AlexNet, the VGG architectures employ ReLUs for response normalisation, and the activations of the

[3]For more information regarding the activation functions, the interested reader is referred to Section 2.1.4.



(a) Cividis (b) Magma

(c) Plasma (d) Viridis

Figure 4.3: A mel-spectrogram plot of an audio sample from an individual snoring (file id: train_0043.wav) from the MPSSC dataset [143] with four different colour maps, *cividis*, *magma*, *plasma*, and *viridis*. The colour bar to the right shows the colour changes associated with increasing spectral energy.

fully connected layers can be obtained to form the deep representations.

4.2.2.3 GoogLeNet

In contrast to AlexNet and VGG networks, GoogLeNet uses inception modules in succession (cf. Figure 4.4). This module consists of a number of parallel convolutional layers and a max-pooling layer. The outputs of all layers are concatenated to produce a single output. The inception module thus collects multi-level features from every input on different scales. The DEEP SPECTRUM features are extracted from the activations of the last pooling layer.

Table 4.1: Comparison between three pre-trained CNNs, AlexNet, VGG16, and VGG19, for extracting DEEP SPECTRUM features. *ch* stands for channels and *conv* denotes convolutional layers.

AlexNet	**VGG16**	**VGG19**
input: RGB image		
1 conv layer size: 11; ch: 96; stride: 4	2 conv layer size: 3; ch: 64; stride: 1	
maxpooling		
1 conv layer size: 5; ch: 256	2 conv layer size: 3; ch: 128	
maxpooling		
1 conv layer size: 3; ch: 384	3 conv layer size: 3; ch: 256	4 conv layer size: 3; ch: 256
	maxpooling	
1 conv layer size: 3; ch: 384	3 conv layer size: 3; ch: 512	4 conv layer size: 3; ch: 512
	maxpooling	
1 conv layer size: 3; ch: 256	3 conv layer size: 3; ch: 512	4 conv layer size: 3; ch: 512
maxpooling		
fully connected layer *fc6*, 4 096 neurons		
fully connected layer *fc7*, 4 096 neurons		
fully connected layer, 1 000 neurons		
output: soft-max of probabilities for 1 000 object classes		

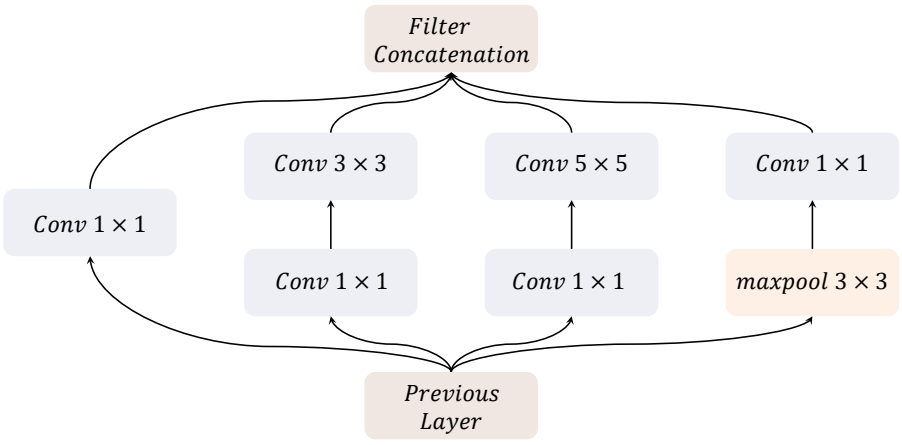


Figure 4.4: An inception module used in the GoogLeNet architecture. Small 1×1 convolutions are applied to reduce the dimensionality. To combine information found at different scales, filters of different path sizes are concatenated.

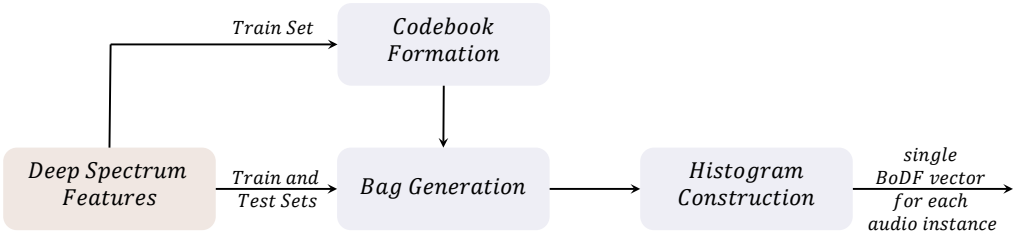


Figure 4.5: Generating BoDF from high-dimensional Deep Spectrum representations.

4.2.3 Bag-of-Deep-Features

The author and his colleagues proposed BoDF, which is a quantised representation of high-dimensional Deep Spectrum features [88]. The main goal for the quantisation step is to compress the feature space and increase its to noise related adverse and confounding effects.

In order to form BoDF, fixed length histogram representations of the time-continuous Deep Spectrum features are generated using openXBOW[4], an open-source toolkit for the generation of Bag-of-Words representations [145]. This is achieved by first identifying a set of 'deep audio words' from given training data, and then bagging the original feature space, with respect to the generated codebook, to form the histogram representation. The histogram shows the frequency of each identified deep audio word in a given audio instance [145–148]. It is worth noting that the audio words do not represent words in their semantic meaning, but rather fragments of the audio signal defined by features [146]. The codebook can be the result of, e. g. a clustering algorithm [149], or a random sampling[5] of low-level descriptors [150]. The histogram finally describes the distribution of the codebook vectors over the whole audio segment [145]. The overall structure of the quantisation process is depicted in Figure 4.5.

In [88], the efficacy of BoDF has been demonstrated for soundscape classification of audio recorded in real-world environments. The dataset for the experiments in [88] have been sourced from YouTube with *Cost-efficient Audio-visual Acquisition via Social-media Small-world Targeting* (CAS^{2}T) toolkit for efficient large-scale big data collection [151]. However, the main drawback of using BoDF is that it does not consider the temporal dependencies of the audio signals (e. g. continuous speech), as the frame-level Deep Spectrum feature vectors for the time-series are mixed up to build a clip-level feature vector.

[4] `https://github.com/openXBOW/openXBOW`

[5] The sampling step in openXBOW is done with a defined random seed.

5

Deep Convolutional Generative Adversarial Networks

Deep Convolutional Generative Adversarial Networks (DCGANs) are a type of GANs which use CNNs in the generator and discriminator (for more detailed information regarding GANs, the reader is referred to Section 3.2). Both the generator and discriminator consist of convolutional layers, but unlike CNNs they do not have any pooling layers. DCGANs have been first introduced by Radford et al. [48] and have been applied for image classification tasks with state-of-the-art results [48]. In this chapter, the methodology applied by the author and his colleagues to utilise DCGANs for unsupervised representation learning from audio data will be described [152].

5.1 Characteristics

For the task of unsupervised representation learning for acoustic scene classification [1], the author and his colleagues have introduced a DCGAN architecture (cf. Section 5.2) based on the structure and results reported by Radford et al. [48, 152].

The DCGAN structure proposed by Radford et al. [48] adheres to various limitations. First, all pooling layers are replaced with fractional-strided convolutions for the generator and strided convolutions for the discriminator [153]. Second, *Batch Normalisation* (BN) is applied in all layers of both the generator and the discriminator, except for the last layer of the generator and the first layer of the discriminator. These two layers are excluded, in order to allow the model to learn the correct mean and scale of the data distribution [41, 48]. Third, no fully connected layers are included on top of the convolutional layers. Fourth, a ReLU activation is applied in the hidden layers of the generator, and leaky ReLU is used for the hidden layers of the discriminator [47]. Finally, hyperbolic tangent and softmax activations are used for the output layer of the generator and discriminator, respectively.

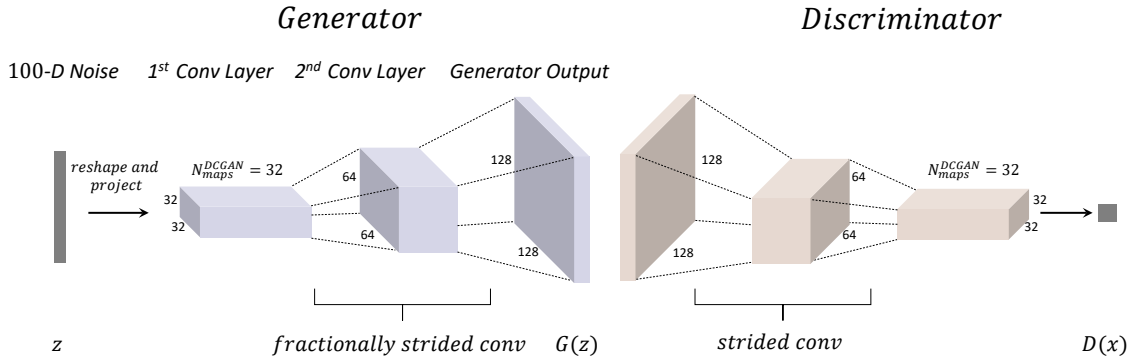


Figure 5.1: Illustration of the DCGAN architecture with $N_{layer}^{DCGAN} = 2$ and $N_{maps}^{DCGAN} = 32$ applied for generating spectrograms with hypothetical dimensions 128×128. A 100 dimensional Gaussian noise is projected and reshaped to a spatial convolutional representation. In every convolutional layer below the output layer, the spatial representations are halved. The convolutional layer directly below the output layer has N_{maps}^{DCGAN} feature maps. The number of the feature maps is then doubled in each further layer. The discriminator mirrors the CNN architecture of the generator. The number of convolutional layers in both generator and discriminator is equal.

5.2 Architecture

The DCGANs architecture applied by the author and his colleagues [152] has a less complex architecture ($N_{layer}^{DCGAN} = 2$ and $N_{maps}^{DCGAN} = 32$) than the DCGAN proposed by Radford et al. [48], who use $N_{layer}^{DCGAN} = 4$ and $N_{maps}^{DCGAN} = 64$. N_{layer}^{DCGAN} denotes the number of convolutional layers in the generator and discriminator CNNs, and N_{maps}^{DCGAN} is the number of the feature maps in the output layer of the generator.

This simpler structure for audio analysis has been applied for three main reasons. First, the amount of audio data employed for the DCGAN experiment in this thesis (cf. Section 8.1.1) is much less than the image data used in Radford et al.'s experiments [48]. Second, for the decreased amount of training data, the number of free parameters is reduced. Third, in [152], grey-scale spectrograms with one channel have been used, whilst Radford et al. train their DCGANs on colour images with three channels.

For the introduced DCGAN (cf. Figure 5.1), both the generator and discriminator comprise of an equal number N_{layer}^{DCGAN} of convolutional layers with a fixed stride of two. The output layer of the generator and the input layer of the discriminator have the spatial dimensions of the input spectrograms that should be processed and contain N_{maps}^{DCGAN} feature maps. In each layer, directly below the layer in the generator or on top of the layer in the discriminator, the number of the feature maps is doubled and the spatial dimensions are halved. A 100-dimensional uniform noise distribution is applied as input to the generator, where it is projected to the dimensionality required by the first convolutional layer. The discriminator mirrors the architectural properties of the generator and therefore it has

Table 5.1: Mathematical symbol of the hyperparameters for the DCGAN, and the range of each hyperparameter.

Symbol	Range	Description
N_{layer}^{DCGAN}	$\mathbb{N}^+$	Number of convolutional layers in the generator and discriminator CNNs
N_{maps}^{DCGAN}	$\mathbb{N}^+$	Number of feature maps in the output layer of the generator, and the input layer of the discriminator

an input layer with spatial dimensions of the spectrogram [152].

Feature matching has been introduced as an effective technique to improve GAN stability [154]. Using feature matching, the generator is trained to produce activations in an intermediate layer of the discriminator that are similar to those produced by the real training examples. In [152], the author and his colleagues applied this training approach instead of the conventional objective function for the generator. After the training process is finished, the activations of the convolutional layers in the discriminator are obtained as the deep representation of an input spectrogram. To reduce the dimensionality of this representation, the activations of each feature map are max-pooled onto a 4×4 spatial grid and then flattened and concatenated to form a single vector [48]. This final representation can then be directly applied for training of a classifier.

6

Recurrent Sequence-to-Sequence Autoencoders

Acoustic sequences are typically varying length signals; this highlights a major drawback for CNNs-based representation learning methodologies introduced in Chapters 4 and 5, which generally require inputs of fixed dimensionality. Whilst these networks are very effective to learn high-level local structures from the spectrograms, their ability to learn the long-term temporal context from input audio signals is limited [155, 156]. Moreover, many DNN systems applied for representation learning, including RBMs or *stacked autoencoders*, do not explicitly account for the inherent sequential nature of audio signals [10]. Sequence to sequence learning with RNNs has been first proposed in machine translation [44, 157, 158]. S2SAEs have been used for unsupervised pre-training of RNNs with state-of-the-art results on image recognition or text classification tasks [159]. Variational autoencoders have been employed to learn representations of sentences and to create new sentences from the latent space [160, 161]. Furthermore, Weninger et al. used denoising recurrent autoencoders to learn variable-length representations of audio for reverberated speech recognition [162].

In this chapter, the important characteristics and the structure of the RNN-based S2SAE proposed by the author and his colleagues are introduced [17, 18]. This methodology is developed to learn fixed-length representations from variable-length audio data with sequential nature. This approach has shown its strength in various audio recognition tasks (cf. Sections 8.1 and 10.1) [152, 163] and has been applied as a baseline system for the well-known 2018 edition of the INTERSPEECH *Computational Paralinguistics ChallengE* (COMPARE) [164].

6.1 Characteristics

The developed S2SAE is built of LSTM [43] cells or GRUs [44], and can be directly trained on spectrograms, which are viewed as time-dependent sequences of frequency

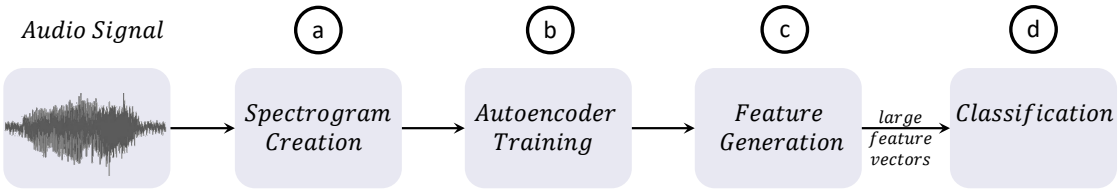


Figure 6.1: High-level structure of the proposed S2SAE for deep representation learning. The autoencoder training is entirely unsupervised.

vectors. Fully unsupervised autoencoder training, and the ability to account for the temporal dynamics of input sequences are two of the key strengths of the proposed S2SAE. The author and his colleagues have also published AUDEEP[1], an open-source Python toolkit for unsupervised feature learning utilising the proposed S2SAE [17, 18]. A step-by-step tutorial on how to perform representation learning with AUDEEP, and a complete documentation of the AUDEEP command line interface is provided on the GitHub repository page.

6.2 Architecture

A high-level overview of the proposed deep representation learning system using S2SAE is given in Figure 6.1. First, visual representations are generated from audio signals (cf. Figure 6.1a). A similar procedure for audio plots creation, as described in Section 4.2.1 is applied. A sequence to sequence autoencoder is then trained on the generated audio plots (cf. Figure 6.1b). Afterwards, the learnt representation of each input instance is extracted as its feature vector (cf. Figure 6.1c). Finally, if instance labels are available, a classifier can be trained and evaluated on the obtained features (cf. Figure 6.1d).

The structure of the S2SAE is based on the recurrent encoder-decoder model first introduced for machine translation [44, 157]. A high-level architecture of the autoencoder is depicted in Figure 6.2. Audio plots (cf. Section 4.2.1) are considered as a time-dependent sequence of frequency vectors that describe the power spectral density in the frequency bands across the audio plots' frames. The number of frames in an audio plot depends on the length of the input audio sample, therefore these sequences may have varying length.

The process of reconstructing of the input sequences is done in three steps. First, the input frequency vector sequence is fed to a multilayered *encoder* RNN that updates its hidden state in each time step based on the input frequency vector. Hence, the final hidden state of the encoder RNN contains information about the entire input sequence, and can be considered as a fixed-length representation of the input audio plot. This representation, i. e. the final hidden state of the encoder RNN, is then forwarded across a fully connected layer. Afterwards, a multilayered *decoder* RNN aims to reconstruct the input sequence

[1] `https://github.com/auDeep/auDeep`

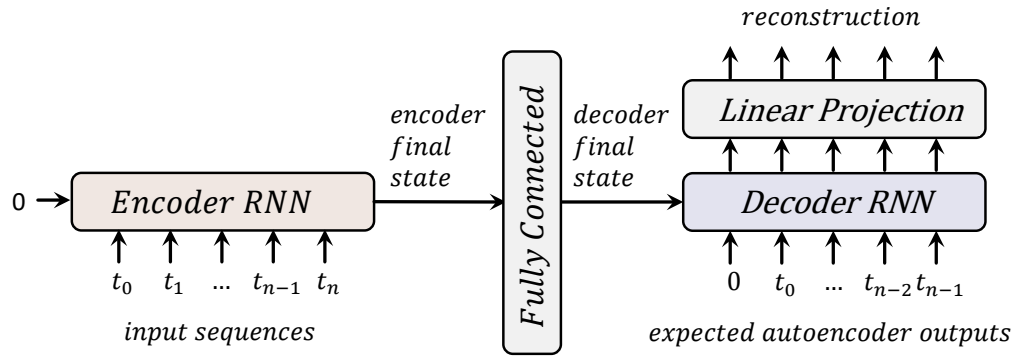


Figure 6.2: A high-level overview of the recurrent autoencoder utilised in the proposed S2SAE. The activations of the fully connected layer between the decoder and encoder units are extracted as the learnt representations.

Table 6.1: Mathematical symbol of the hyperparameters for the S2SAE, and the range of each hyperparameter.

Symbol	Range	Description
N_{layer}^{S2SAE}	$\mathbb{N}$	Number of recurrent layers in the encoder and decoder RNNs
N_{unit}^{S2SAE}	$\mathbb{N}$	Number of recurrent units in each encoder and decoder RNNs layer
T_{cell}	$\{LSTM, GRU\}$	Type of cells in the encoder and decoder RNNs
D_{enc}	$\{B, U\}$	Direction of the encoder (B: bidirectional, U: unidirectional)
D_{dec}	$\{B, U\}$	Direction of the decoder (B: bidirectional, U: unidirectional)

based on the information contained in the transformed representation and the decoder RNN input.

The encoder RNN has N_{layer} recurrent layers, each of which contains equal number N_{unit} of uni- ($D_{enc} = U$) or bidirectional ($D_{enc} = B$) GRU [44] or LSTM [43] recurrent cells. The input audio plots are normalised between $[-1, 1]$ before being fed to the encoder RNN, and the hidden state of each recurrent cell is initialised with zeros. The final hidden states of the cells are concatenated into one vector. This vector is then forwarded across a fully connected layer with hyperbolic tangent activation (the activation functions used in this thesis are described in Section 2.1.4).

In the final step, the decoder RNN has the same number of recurrent layers and units as the encoder RNN. However, it is possible to choose its direction (D_{dec}) regardless of the encoder RNN. The task of the decoder RNN is a frame-wise reconstruction the input spectrogram. A linear projection layer (cf. Figure 6.2) with weights shared throughout time steps is applied after the decoder RNN, in order to project its outputs to the required dimensions. The reason behind using the projection layer is to match the output dimen-

sions of the decoder RNN and the spectrogram frequency vectors.

During the training phase, the network attempts to minimise the *Root Mean Squared Error* (RMSE) between the decoder RNN output and the target sequence. Dropout is applied to the inputs and outputs of the recurrent layers, but not to the hidden or fully connected layers [38]. After the autoencoder training is finished, the activations of the fully connected layer are obtained as the deep representation of the input sequence. The number of the features (N_{feat}^{S2SAE}) for each input spectrogram after the training process is:

$$N_{feat}^{S2SAE} = \begin{cases} N_{layer}^{S2SAE} \cdot N_{unit}^{S2SAE}, & \text{if } D_{dec} = U \text{ and } T_{cell} = GRU \\ 2 \cdot N_{layer}^{S2SAE} \cdot N_{unit}^{S2SAE}, & \text{if } D_{dec} = B \text{ and } T_{cell} = GRU \\ & \text{or } D_{dec} = U \text{ and } T_{cell} = LSTM \\ 4 \cdot N_{layer}^{S2SAE} \cdot N_{unit}^{S2SAE}, & \text{if } D_{dec} = B \text{ and } T_{cell} = LSTM \end{cases}$$

The representations learnt by the S2SAE can be extensive, containing several thousands of dimensions. Whilst this is still lower than the dimensionality of raw audio signals, a classification algorithm may struggle to process such large feature vectors [165]. There is also a linear relationship between the number of the learnt representations and the number of the hidden layers and recurrent units. Increasing each of these factors can quickly lead to issues with training time and the 'curse of dimensionality', which refers to the challenge of finding structure in data embedded in a high dimensional space [166]. Hence, it is recommended to apply dimensionality reduction methods to the obtained representations before forwarding them to a classifier.

The author and his colleagues have performed a comparative study of the following archetypal feature selection approaches for large feature spaces [165]: (i) a *Principal Component Analysis* (PCA) [167], (ii) a filter-based feature selection using *Canonical Correlation Analysis* (CCA) [168], (iii) a *Correlation-based Feature Selection* (CFS) [169], and (iv) a wrapper-based feature selection with *Sequential Forward Selection* (SFS) [170], and a *Competitive Swarm Optimisation* (CSO) [171–173].

As the detailed comparison between the mentioned dimensionality reduction methods is beyond the scope of this thesis, the interested reader is referred to [165] or the corresponding references of each method.

7

Convolutional Recurrent Neural Networks

Uniquely, the CNN-based and S2SAE methodologies proposed in Chapters 4 to 6 are individually able to learn hierarchically robust visual features, and the long-term temporal context from the input audio. However, these methods are not able to observe both of these representations simultaneously. This problem is addressed in this chapter by introducing a deep *Convolutional Recurrent Neural Network* (CRNN) approach for audio processing.

7.1 Characteristics

A CRNN, which is a combination of a CNN and an RNN, has shown to be able to learn local dependencies of the input signal by adopting convolutional layers, whilst obtaining the global structures with recurrent layers [104, 106, 107]. Based on the structure and the results provided by Lim et al. [116] and Cakir et al. [117, 119], the author and his colleagues have proposed a CRNN utilising LSTM cells with backward pass in the recurrent layers for detection of rare acoustic events [128] and a CRNN with BLSTM cells in the recurrent layers for classification of different, rare vocalisations from children with an ASC [127]. In the following section, the architecture of the introduced CRNN is given.

7.2 Architecture

Motivated by the systems presented in both [116, 117], the author and his colleagues implemented a CRNN composed of four main components. First, log mel-spectrograms are extracted from the audio recordings (cf. Figure 7.1a). These samples are then fed as input into the convolutional layers to extract high-level spectral features (cf. Figure 7.1b). Afterwards, recurrent layers are applied to learn the long-term temporal context from the obtained features (cf. Figure 7.1c). Finally, a FFNN with softmax activation is used to classify the input data (cf. Figure 7.1d).

Figure 7.1: Illustration of the proposed CRNN approach composed of convolutional and recurrent neural networks for feature extraction and a feed forward network to generate the final predictions. A detailed account of the procedure is given in Section 7.2.

Table 7.1: Mathematical symbol of the hyperparameters for the CRNN, and the range of each hyperparameter.

Symbol	Range	Description
N_{layer}^{CRNN}	$\mathbb{N}$	Total number of layers in the CRNN, including CNN, RNN, and feed-forward layers
N_{unit}^{RNN}	$\mathbb{N}$	Number of recurrent units in each RNN layer
N_{unit}^{FFNN}	$\mathbb{N}$	Number of units in the fully connected layer

7.2.1 Creation of Log Mel-Spectrograms

Several studies have shown the effectiveness of log mel-filterbank energy coefficients for speech recognition [174,175], and *Sound Event Detection* (SED) [116,117,176]. A major advantage borne by these features in comparison to simple spectrograms is the filtering of frequency components with log scale filterbanks, which is based on the frequency response of the human ear that has better resolution at lower frequencies.

For the introduced CRNN, mel-filterbanks are extracted as following. First, the mel-spectrograms are divided into frames of width w ms and overlap of $0.5\,w$ ms from the log-magnitude spectrum by dimensionality reduction using a mel-filter. N_{mel} mel-filterbanks are then applied equally spaced on the mel-scale. Afterwards, log operation is taken on the power spectrograms. The reason behind this step is that the perceptual loudness of an audio signal is approximately logarithmic [177]. Finally, the log mel-spectrograms are then divided into chunks of a desired time step τ. At this stage, this visual representation of the input audio signal is ready to be fed into the CNN (cf. Section 7.2.2).

7.2.2 Convolutional Layers

The log mel-spectrograms are fed into a convolutional layer with 2D filters. As depicted in Figure 7.2, the frequency time convolution is followed by non-overlapping pooling to ensure no shrinking in time. Subsequently, a 1D convolution along the spectral domain is applied, which is then followed by max pooling along the frequency domain. ReLU

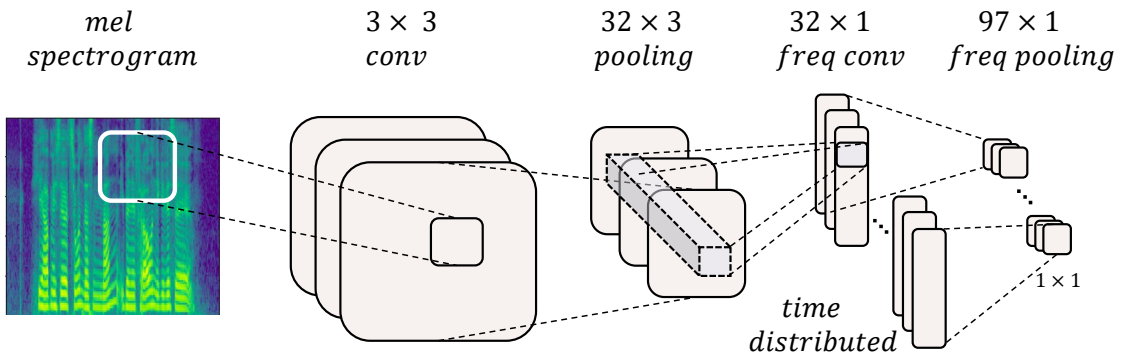


Figure 7.2: The structure of the 2D CNN applied for extracting high-level features from the input log mel-spectrograms.

activation is used in the convolutional layers [45], and BN [41] is applied between them. Furthermore, a dropout with a probability of p can be applied to all layers to add regularisation and also to minimise potential overfitting problems caused by non-overlapping max pooling [38]. The convolutional layers, however, are capable of effectively capturing short-term temporal context. In order to achieve longer temporal modelling, the outputs from the CNN are forwarded to an RNN with either LSTM or BLSTM cells.

7.2.3 Recurrent Layers

The activations emerging from the CNN are passed to a network comprising of two RNN layers. Each RNN layer consists of N_{unit}^{RNN} hidden units with either LSTM or BLSTM cells. Hyperbolic tangent has been used as the activation function, and a dropout with a probability of p can be applied to each layer for regularisation. In the final step, RNN features are extracted for each time step, which are passed on to the fully connected layer to obtain prediction results. The structure of the BLSTM-RNN applied in the CRNN approach for classification of ASC vocalisations is given in Figure 7.3 [127].

7.2.4 Fully Connected Layer

The features returned for each time step from the RNN layers are fed into a fully connected FFNN comprising of a single fully connected layer with N_{unit}^{FFNN} hidden units, matching the depth of the input features. BN is applied to the output, so that the mean is close to 0 and standard deviation is close to 1. The activation function used is ReLU, which adds the desired non-linearity to activations. The updated features are further fed into the output layer to obtain the predictions.

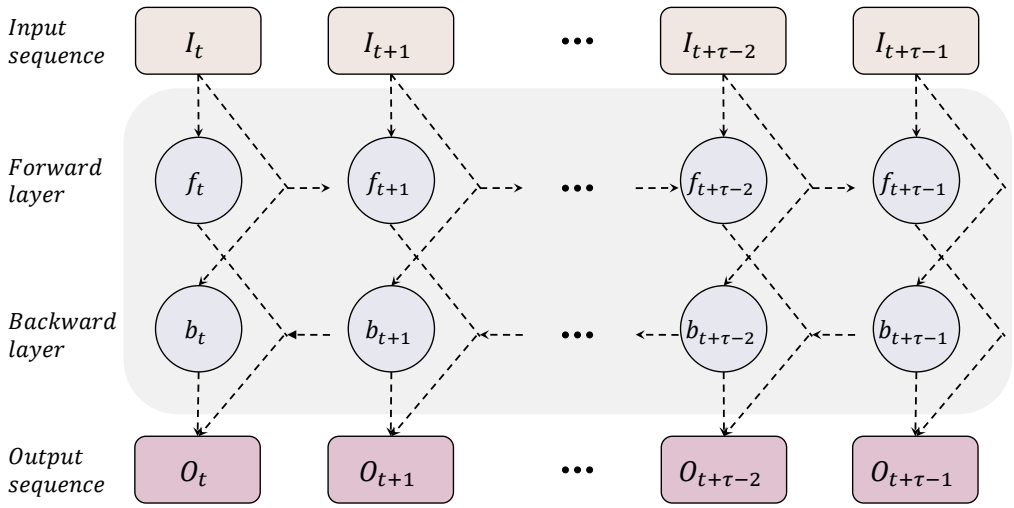


Figure 7.3: The BLSTM-RNN structure applied in the CRNN approach. Two hidden layers, one in the forward direction (f) and another one in the backward direction (b) are used. For all input CNN features ($I_{t+\tau}$) during the time step (τ), the returned outputs from each layer are then concatenated ($O_{t+\tau}$).

Part IV

Experiments

8

Acoustic Sounds and Game Audio

In this chapter, applications of the introduced deep representation learning techniques for problems, such as acoustic scene classification (cf. Section 8.1), rare acoustic event detection (cf. Section 8.2), and audio-based game genre classification (cf. Section 8.3), will be discussed. First, in Section 8.1, a novel combination of features learnt using both a DCGAN (cf. Chapter 5) and a recurrent S2SAE (cf. Chapter 6) will be introduced. Each of the representation learning algorithms is trained individually on spectral features extracted from acoustic sounds. This system is evaluated on the *TUT Acoustic Scenes 2017* dataset [1]. Afterwards, in Section 8.2, the problem of rare acoustic event detection will be addressed. A CRNN approach will be applied and evaluated on the *TUT Rare Sound Events 2017* [1]. Finally, in Section 8.3, the DEEP SPECTRUM system (cf. Chapter 4) will be utilised to solve an audio-based game genre classification problem. For this task, the author and his colleagues have introduced a new database containing 1 566 recordings of 300 individual games from 6 different genres.

8.1 Acoustic Scene Classification

Machine learning algorithms for audio processing typically operate on expert-designed feature sets extracted from the raw audio signals. Arguably among the most widely used features are mel-band energies and features derived from them, such as *Mel-Frequency Cepstral Coefficients* (MFCCs). Both feature spaces are widely used in acoustic scene classification [178–180], with the former being employed in the *Detection and Classification of Acoustic Scenes and Events* (DCASE) 2017 Challenge baseline system [1], and the later being the low level feature space used by the winners of the DCASE 2016 acoustic scene classification challenge [181]. In this section, the DCGAN and S2SAE approaches introduced in Chapters 5 and 6 have been used for deep unsupervised representation learning from acoustic data [18].

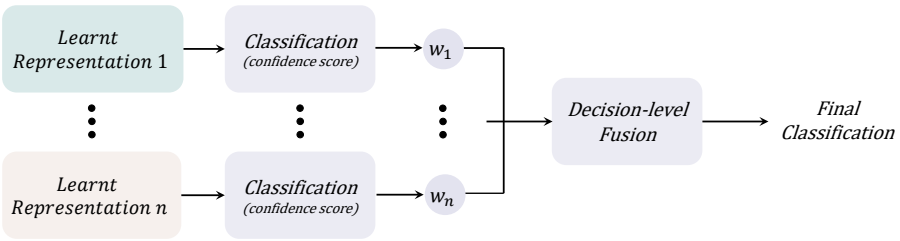


Figure 8.1: An overview of the weighted decision-level fusion system of n learnt representation vectors for an audio recording. A detailed description of the fusion method is given in Section 8.1.1.

8.1.1 Data and Procedure

The proposed deep learning system is composed of two components for unsupervised representation learning: i) a DCGAN, and 2) a S2SAE. First, the activations of the discriminator for the DCGAN and the activations of the fully connected layer between the decoder and encoder units of the S2SAE are extracted (cf. Section 8.1.1.1). As shown in Figure 8.1, separate classifiers are trained on the individual feature sets, and the resulting prediction probabilities are fused [152]. For the decision-level fusion, the predictions and confidence scores are first calculated individually on the DCGAN and S2SAE representations. Afterwards, the decisions are weighted regarding to predetermined weights and fused to obtain the final classification results. The weights for n representations are optimised in the following manner. First, all combination of weights $w_1, \ldots, w_n \in [0,1]$ with $\sum_{i=1}^{n} w_i = 1$ are tested in steps of 0.1. For example, for two representations the weights combinations for (w_1, w_2) would be: $(0.0,1.0),(0.1,0.9),(0.2,0.8),\ldots,(1.0,0.0)$. Afterwards, the weights with which the highest classification result can be achieved are selected.

8.1.1.1 Representation Learning

First, as introduced in Section 4.2.1, mel-spectrograms are extracted from raw audio files. The challenge corpus (cf. Section 8.1.1.2) contains audio samples which have been recorded in stereo [1]. In such datasets, there may be instances in which important information related to the class label has been captured in only one of the two channels. Following the winners of the DCASE 2016 acoustic scene classification challenge [181], mel-spectrograms are extracted from each individual channel, as well as from the mean and difference of the two channels. Separate sets of mel-spectrograms are extracted for different parameter combinations, each containing one mel-spectrogram per audio sample. Representations are then learnt independently on different sets of mel-spectrograms. Afterwards, a DCGAN and recurrent S2SAE are trained on these spectra. After DCGAN and autoencoder training, the learnt representations of the mel-spectrograms are extracted to be used as feature vectors for the corresponding instances. This step is repeated for the

obtained mel-spectrograms from each audio channel (left and right), and the mean and difference between them.

8.1.1.2 Database

As mentioned, the proposed system is evaluated on the *TUT Acoustic Scenes 2017* dataset from the DCASE 2017 acoustic scene classification challenge [1]. This dataset contains binaural audio samples from 15 acoustic scenes recorded at distinct geographical locations. For each location, between 3 and 5 minutes of audio were recorded and then split into 10 second segments. The development set for the challenge contains 4 680 instances, with 312 instances per class, and the evaluation set contains 1 620 instances.

A four-fold CV setup is provided by the challenge organisers for the development set. In each fold, roughly 75 % of the samples are used as the training split, and the remaining samples are used as the evaluation split. Samples from the same original recording are always included in the same split. The experiments are conducted on the development set only, as there is a remarkable mismatch in recording conditions between the partitions which causes observable confounding effects [152]. This is evidenced by the lack of relationship between the development and evaluation scores in the 2017 challenge[1]. For further details on the challenge data and the CV setup, the interested reader is referred to [1].

8.1.1.3 Multilayer Perceptron Classifier

An MLP with two hidden fully connected layers ($N_{layer}^{MLP} = 2$) with ReLU activation, and a softmax output layer is applied for classification. The hidden layers contain 150 units each ($N_{unit}^{MLP} = 150$), and the output layer contains one unit for each class label (i. e. 15 neurons for all 15 classes). Training is performed using cross entropy between the ground truth and the network output as the objective function, with dropout applied to all layers except the output layer [38].

8.1.1.4 Common Experimental Settings

The S2SAEs, DCGANs, and MLP are trained using the Adam optimiser [182]. Autoencoders are trained for 50 epochs in batches of 64 samples with an initial learning rate of 0.001, and 20 % dropout is applied to the outputs of each recurrent layer. Furthermore, gradients with absolute value above 2 are clipped [157]. The DCGANs are trained for 10 epochs in batches of 32 examples, and an initial learning rate of 0.0002 and momentum $\beta_1 = 0.5$ is applied. The MLPs used for classification are trained for 400 epochs without batching or gradient clipping, and 40 % dropout is applied to the hidden layers.

[1] `http://www.cs.tut.fi/sgn/arg/dcase2017/challenge/task-acoustic-scene-classification-results`

Both the autoencoders and MLPs are trained using the Adam optimiser with an initial learning rate of 0.001 [182]. Autoencoders are trained for 50 epochs in batches of 64 samples, and 20 % dropout is applied to the outputs of each recurrent layer. Furthermore, gradients with absolute value above 2 [157] are clipped. The MLPs used for classification are trained for 400 epochs without batching or gradient clipping, and 40 % dropout is applied to the hidden layers. Features are standardised to have zero mean and unit variance during MLP training, and the corresponding coefficients are used to transform the validation data.

8.1.1.5 Hyperparameter Optimisation

Both representation learning systems contain a wide range of adjustable hyperparameters, which prohibits an exhaustive analysis of the parameter space. For DCGAN, previous work is consulted extensively to guide parameter selection [48]. Based on the results reported by Radford et al. [48], who use $N_{layer}^{DCGAN} = 4$ and $N_{maps}^{DCGAN} = 64$, a less complex DCGAN architecture with $N_{layer}^{DCGAN} = 2$ and $N_{maps}^{DCGAN} = 32$ is applied in this section.

For the S2SAE, suitable values for the hyperparameters are selected in stages, using the results of the preliminary experiments to bootstrap the process. During these experiments, it has been observed that very similar parameter choices lead to comparable performance on spectrograms extracted from different combinations of the audio channels (mean, difference, left and right). Therefore, hyperparameter optimisation is performed on the mean-spectrograms only, and the resulting parameters are used for other spectrogram types [18].

In the first development stage, a suitable autoencoder configuration is selected, i. e. the optimal number of recurrent layers N_{layer}^{S2SAE}, the number of units per layer N_{unit}^{S2SAE}, and either unidirectional or bidirectional encoder and decoder RNNs. In this phase, autoencoders are trained on mel-spectrograms extracted with window width $w = 160$ ms, window overlap 80 ms, and $N_{mel} = 320$ mel-frequency bands, without amplitude clipping. $N_{layer}^{S2SAE} \in \{1,2,3\}$, $N_{unit}^{S2SAE} \in \{16,32,64,128,256,512\}$, and all combinations of uni- or bidirectional encoder and decoder RNNs are evaluated. The highest classification accuracy was achieved when using $N_{layer}^{S2SAE} = 2$ layers with $N_{unit}^{S2SAE} = 256$ units, a unidirectional encoder RNN, and a bidirectional decoder RNN [18].

The second stage served to optimise the window width w, which was used for the mel-spectrogram extraction. The autoencoder configuration determined in the first stage is applied, and set $N_{mel} = 320$. The window width w is evaluated between 0.04 and 360 ms in steps of 40 ms. For each value of w, the window overlap is chosen to be $0.5\,w$. As shown in Figure 8.2a, classification accuracy quickly rises above 84 % for $w > 100$ ms, and peaks at 85.0 % for $w = 200$ ms and $w = 280$ ms. For larger values of w, classification accuracy drops again. As a larger window width may blur some of the short-term dynamics of the audio signals, $w = 200$ ms is chosen. Accordingly, the window overlap is set to be 100 ms [18].

In the third and final optimisation stage, various numbers of mel-frequency bands

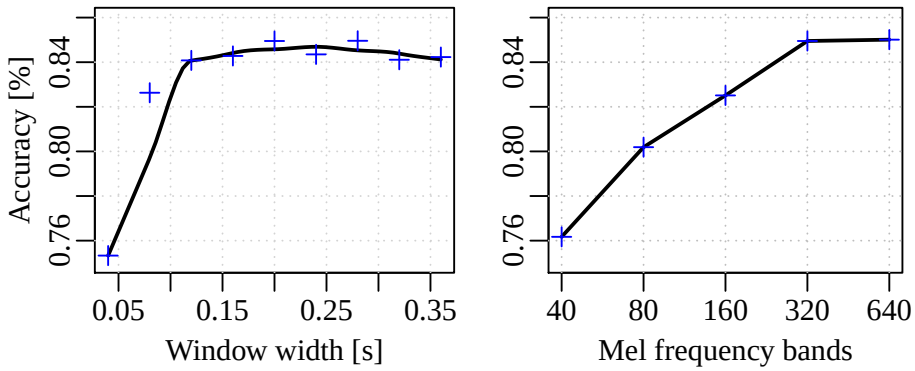


Figure 8.2: Classification accuracy on the development set for different FFT window widths (a), and different numbers of mel-frequency bands (b). In Section 8.1.1.5, a detailed description of the experiments leading to these results is given.

$N_{mel} \in \{40, 80, 160, 320, 640\}$ are tested, the results of which are shown in Figure 8.2b. Classification accuracy rises with larger values of N_{mel} until it reaches 85.0 % for $N_{mel} = 320$. Increasing N_{mel} beyond 320 does not improve performance further. Therefore, $N_{mel} = 320$ is chosen to reduce the computational time [18].

8.1.2 Results

Four sets of spectrograms are extracted from the mean and difference of channels, and from the left and right channels individually [152]. On each set of spectrograms, a DCGAN and a S2SAE are trained, and the learnt representations are extracted as features for the audio instances. This results in four individual feature sets for each approach, herein identified by the spectrogram type from which they have been obtained (i. e. 'mean', 'difference', 'left', and 'right'). The highest individual classification accuracies are achieved from the 'right' feature set (84.5 %) for the DCGAN and the 'mean' feature set (86.0 %) for the S2SAE (cf. Table 8.9).

For each of the 8 individual feature sets (4 for DCGAN and 4 for S2SAE), a classifier is trained and the resulting prediction probabilities are fused in two steps. First, the results on DCGAN-features and S2SAE-features are combined with separately optimised weights. The resulting prediction probabilities (one for DCGAN and one for S2SAE) are then fused with optimised weights [152].

In the first fusion step, accuracies of 86.4 % and 88.5 % are achieved on the fused DCGAN and S2SAE predictions, respectively (cf. Table 8.9). In the final step, the highest classification accuracy of 91.1 % is obtained on the fused prediction probabilities of DCGAN and S2SAE [152].

Table 8.1: Comparison of the classification results of the proposed DCGAN and S2SAE systems with the challenge baseline. Four different feature sets of spectrograms are extracted from the mean (M) and difference (D) of channels, and from the left (L) and right (R) channels separately. The highest accuracy is obtained after fusing the prediction probabilities of DCGAN and S2SAE. CV: Cross Validation.

System	Features	CV Accuracy [%]
Baseline [1]	200 (per frame)	74.8
Proposed: DCGAN		
Mean (M)	3 072	84.1
Left (L)	3 072	83.4
Right (R)	3 072	84.5
Difference (D)	3 072	83.5
Fused (M + L + R + D)		86.4
Proposed: S2SAE		
Mean (M)	1 024	86.0
Left (L)	1 024	84.9
Right (R)	1 024	84.0
Difference (D)	1 024	82.0
Fused (M + L + R + D)		88.5
Proposed: DCGAN + S2SAE		**91.1**

8.1.3 Conclusions

This section analysed the effectiveness of using deep unsupervised representation learning algorithms for the task of acoustic scene classification. In this regard, a novel combination of features generated using a DCGAN and a S2SAE was proposed. Results presented indicate that fusing the prediction probabilities of each classifier trained on each representation, it is possible to improve upon the challenge baseline of 74.8 % to 91.1 %, representing an improvement of 16.3 percentage points. This result indicates that the two techniques complement each other for the task of acoustic scene recognition. Further, it was demonstrated that adversarial networks learn strong representations from spectral features. Both DCGAN and S2SAE are data driven approaches, and their capability for unsupervised representation learning should be tested on bigger datasets. In the future work, a more extensive parameter search should be conducted. Furthermore, it might be beneficial to conduct early fusion experiments, as the learnt representations might also complement each other.

8.2 Rare Acoustic Event Detection

Monitoring systems using audio microphones, in addition to video sensors, are becoming increasingly popular [183]. Audio is especially useful when video fails to effectively detect an event. The process of detecting an event using the audio modality is described as *Sound Event Detection* (SED). The goal of SED is to recognise individual sounds in audio, including the estimation of onset (beginning of an event) and offset (end of an event) for distinct sound event instances [1]. Real-world audio introduces various challenges for the automatic detection of sound events, such as overlapping of other sounds with the target event, or the variability of acoustic events belonging to the same sound class [1]. A detection system would be faced with additional difficulties if the target sound events are *rare* [1].

Rare Sound Event Detection (RSED), as evidenced by the recent IEEE AASP Challenge on DCASE 2017, is a growing field of acoustic classification research [1]. Such a system has many benefits in surveillance and smart home systems, including intrusion detection based on the sound of glass breaking, or car collision detection. This need has given rise to research interest in developing better techniques for audio event detection, including both monophonic [184] and polyphonic sound events [176, 185]. Monitoring systems need to be able to focus on the specific alarm of interest with high accuracy. Since these sounds rarely occur simultaneously, it is useful to explore monophonic SED techniques for such a task. SED has seen use of *Non-negative Matrix Factorisation* (NMF) [186] for source separation, and *Hidden Markov Models* (HMMs) [184] and *Support Vector Machines* (SVMs) [187] for acoustic modelling. However, recent deep learning approaches, especially CRNNs, have shown to be more effective [64, 114, 176, 188]. Therefore, in this section, the CRNN approach introduced in Chapter 7 is applied for the task of RSED [128].

8.2.1 Data and Procedure

All experiments are performed on the *TUT Rare Sound Events 2017* corpus [1]. The focus of this task is the detection of rare sound events in artificially created mixtures [1]. The experiments rely on the CRNN introduced in Chapter 7. First, log mel-spectrograms are extracted from the input audio recordings with a window width of $w = 46$ ms and overlap $0.5\,w = 23$ ms, and $N_{mel} = 128$ mel-filterbanks are applied to the frequency component of each frame [128]. The log mel-spectrograms are put into chunks with time step τ (e. g. $\tau = 5$ means that 5 short-time spectra each with $w = 46$ ms are put together) to be fed into the convolutional layers with 2D filters. Afterwards, the extracted shift-invariant CNN features are forwarded through the LSTM-RNN layers for temporal modelling. The information flow in the recurrent layers is in a backward direction, and LSTM cells are used. Lim et al. showed that using unidirectional recurrent cells, in which the information is passed in the backward direction, it is possible to achieve a better performance than using other recurrent cells for the mentioned acoustic task [116]. The structure of the

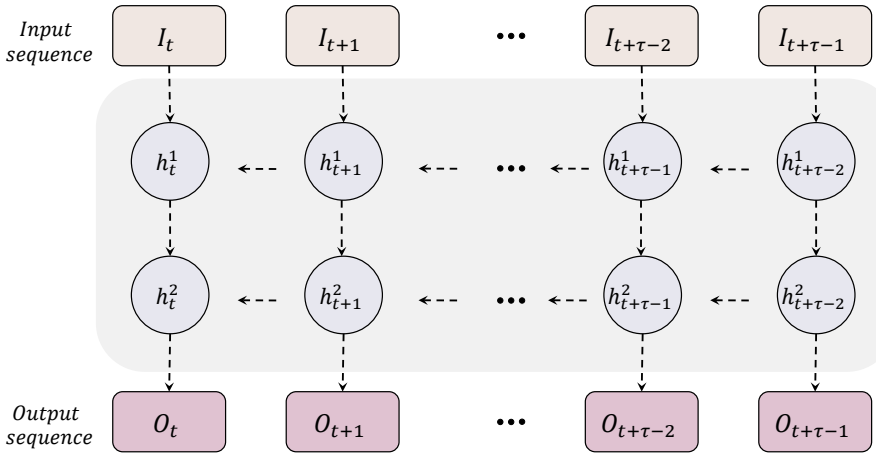


Figure 8.3: Two backward RNN-LSTM layers, each with 128 hidden units ($h_{t+\tau}$). Outputs ($O_{t+\tau}$) are returned for all inputs ($I_{t+\tau}$) during the time step (τ).

LSTM-RNN is given in Figure 8.3. In the final step, a FFNN is used to obtain the final predictions for the acoustic events.

8.2.1.1 TUT Rare Sound Events 2017 Dataset

TUT Rare Sound Events 2017 corpus consists of audio recordings from 15 different everyday acoustic scenes (home, park, train, etc.), some of which are mixed with isolated recordings from one of the three different target sound event classes, namely, *Babycry*, *Glassbreak*, and *Gunshot* [1]. The isolated recordings are divided into segments, and relevant target classes are selected by a human annotator. Mixing is performed by adding a segment to the 30-second long background acoustic scene sample with a random time offset. The normal length of a target event is < 2.5 s, thus enforcing the idea of 'rare' (cf. Table 8.2). There is also a big gap between positive and negative labels in the dataset (cf. Table 8.3).

The following parameters are used for generating the training partition: *Event Presence Probability* (EPP): 0.9, *Mixtures Per Class* (MPC): 1 000, *Event to Background Ratio* (EBR): {-6,0,6} dB. EPP signifies the ratio of the audio clips containing the event to the total number of clips. MPC denotes the total files per each class (with and without events), and EBR stands for event to background ratio.

The total duration of the training set used is 25 hours, which is twice the size of the development set with the parameters EPP: 0.5, MPC: 500, and EBR: {-6,0,6} dB. The evaluation set has the same parameters as that of the development partition, but contains isolated recordings from a different source.

Table 8.2: Statistical data on sound event duration.

Class	Babycry		Glassbreak		Gunshot	
Partition	dev	test	dev	test	dev	test
Mean	2.41 s	1.85 s	1.36 s	0.72 s	1.43 s	1.04 s
Max	5.10 s	4.24 s	4.54 s	1.82 s	4.40 s	3.68 s
Min	0.66 s	0.78 s	0.26 s	0.30 s	0.24 s	0.30 s
Median	2.33 s	1.67 s	1.29 s	0.70 s	1.21 s	0.76 s
SD	0.98 s	0.83 s	0.75 s	0.30 s	0.86 s	0.83 s

Table 8.3: Contrast between the positive and negative labels in the dataset.

Target Event	Positive [%]	Negative [%]
Babycry	4.13	95.87
Glassbreak	2.47	97.53
Gunshot	2.47	97.53

8.2.1.2 Network Hyperparameters

For all the three binary classifiers, similar hyperparameters, except for the batch size and number of epochs, have been applied (cf. Table 8.4). The hyperparameter search is done in the following manner. First, the regularisation is turned off, and Adam optimiser with different learning rates $l_r \in \{0.1, 0.01, 0.001, 0.0001\}$ are tested [182]. Binary cross-entropy is used as loss function. The highest learning rate that decreases the loss in the first few epochs is chosen as the starting learning rate for the network. In the experiments, $l_r = 0.001$ provided the best results. Overfitting is then monitored closely by observing the behaviour difference between training and validation losses. Gradually, regularisation including dropout is introduced to minimise the overfitting problem [38]. A dropout rate of 30 % was shown to be effective. Contrarily, if the model is underfitting, an extra layer is added. The training is performed for a minimum of 100 epochs and then stopped after witnessing no improvement in validation loss for 15 epochs. The hyperparameter settings used in the final configuration are given in Table 8.4.

8.2.1.3 Sliding Ensemble Approach

If an activity is detected for *n* frames, then not detected, for example for two subsequent frames, and afterwards detected again, it is highly possible that the predictions from those two frames are noisy. In order to cope with such short noisy frames, the *sliding ensemble* approach proposed by Lim et al. [116] is applied. The goal of this method is to average the overlapping predictions and obtain smoother outputs [116]. A window size equal to

Table 8.4: Hyperparameters for the CRNNs for each subtask, together with their approach identifiers. N_{layer}^{CNN} : number of convolutional layers; N_{layer}^{RNN} : number of recurrent layers; N_{unit}^{RNN} : number of recurrent units in each RNN layer; N_{unit}^{FFNN} : number of units in the fully connected layer; l_r : learning rate; l_{rd} : learning rate decay.

Parameters	Babycry	Glassbreak	Gunshot
N_{layer}^{CNN}	2	2	2
N_{layer}^{RNN}	2	2	2
N_{unit}^{RNN}	128	128	128
N_{unit}^{FFNN}	128	128	128
l_r	0.001	0.001	0.001
l_{rd}	0.01	0.01	0.01
dropout	0.3	0.3	0.3
epochs	115	118	100
batch size	48	56	64

the number of time steps $\tau > 1$ and with overlap of a time step $\tau = 1$ is used to obtain temporal probability sequence. Fixed thresholding is applied to estimate the presence of an event and the onset time. A threshold of 0.8 for *Babycry* and *Glassbreak*, and 0.5 for *Gunshot* is applied for event presence in an entire audio clip. If an event is present in an audio recording, the peak is then calculated and a certain number of frames is checked before the peak; the first frame with $p > 0.5$ is determined to be the onset [128].

8.2.2 Results

The evaluation of the classification performance is done using event-based *Error Rate* (ER) [189], and can be calculated as follows:

$$ER = \frac{S + D + I}{N} \tag{8.1}$$

where S (*substitutions*) stands for the events in system output with correct temporal position but incorrect class label, I (*insertions*) for the events in system output that are neither correctly detected nor are substitutions (S), D (*deletions*) for the events in ground truth that are neither correct nor substituted, and N for the total number of events in the ground truth [189]. These evaluation metric is computed using the SED-eval-toolbox provided in the DCASE 2017 challenge [1].

Three sets of experiments have been performed to evaluate the robustness of the applied CRNN. First, the effect of using log mel-spectrograms with their first order derivatives (Δ) is analysed (cf. *Experiment 1* in Table 8.6). Second, it is investigated to what

Table 8.5: Final models comprising of weighted average ensembles. t(τ) denotes the number of time steps (τ) used for input chunks.

Target Event	Time Step Ensemble
Babycry	$(t(5)+t(9)+t(50))/3$
Glassbreak	$t(3)$
Gunshot	$(t(3)+t(5))/2$

Table 8.6: The CRNN results on the development partition of the TUT Rare Sound Events 2017 using all three experiments introduced in Section 8.2.2 compared with the baseline system. The best overall result is highlighted with a light grey shading.

	Erro Rate (ER)			
System	Babycry	Glassbreak	Gunshot	*Average*
Baseline [1]	0.67	0.22	0.69	0.53
Experiment 1: Effect of Features				
log mel-spectra + Δ	0.19	0.08	0.26	0.18
Experiment 2: Frame Concatenation t (τ)				
$t(3)$	0.38	0.19	0.41	0.33
$t(5)$	0.36	0.20	0.37	0.31
$t(9)$	0.29	0.18	0.38	0.28
Experiment 3: Sliding Ensemble				
w/o sliding ensemble	0.24	0.17	0.36	0.22
with sliding ensemble	0.18	0.09	0.24	**0.17**

extent frame concatenation can affect the results (cf. *Experiment 2* in Table 8.6). Third, the effect of using sliding ensemble is analysed [116], which combines the probabilities of time steps $t(\tau)$ with overlap of one time step and averages the probabilities (cf. *Experiment 3* in Table 8.6). In Table 8.5, the time step ensembles applied for the target events are given. The final model for each target event uses an average of the weights from the models trained with different time steps τ.

The CRNN results on the development set using all three experiments compared with the baseline system are given in Table 8.6. The best configuration on the development partition is used to obtain the evaluation results, which are shown in Table 8.7.

The results show that for all three sets of experiments, the proposed CRNN outperforms the baseline system by a wide margin on the development partition (cf. Table 8.6). Using the best functioning system on the development set (*Exp3: Sliding Ensemble*), it is

Table 8.7: The CRNN results on the evaluation partition of the TUT Rare Sound Events 2017 using the best performing system on the development set (cf. Table 8.6) compared with a CNN-only approach and the baseline results. The best overall result is highlighted with a light grey shading.

	Erro Rate (ER)			
System	Babycry	Glassbreak	Gunshot	*Average*
Baseline (FFNN) [1]	0.80	0.38	0.53	0.57
CNN with a FFNN	0.48	0.27	0.56	0.43
Experiment 3: Sliding Ensemble				
with sliding ensemble	0.21	0.19	0.41	**0.27**

also possible to outperform the baseline on the evaluation set and achieve strong results for all events (cf. Table 8.7) [128]. The event *Glassbreak* has the best performance regardless of the network configuration. It is most likely because of the nature of the event that the frequency component, at the moment when a glass breaks, is impulsive and distinct in comparison to the background sounds. Therefore, a short time step is effective for this problem. However, sometimes other events with similar onset frequencies get confused. The event *Gunshot* is also an impulsive sound and requires short time step analysis. However, in a *Gunshot* event, there are usually several vibrations between the onset and the offset, hence, relatively longer time step works effectively for this task. In the case of *Babycry*, the event lasts for longer periods and so requires the use of longer time step frames [128].

The improvement achieved by using sliding ensemble during post-processing can be explained by the noise in the posterior probabilities of the event classes, which is obtained from the output layer of the CRNN. There are instances in which the posterior probability has a high peak for a single frame ($w = 46$ ms) compared to the frames before and after. This is not very realistic, since none of the three rare target events occur for only 46 ms and disappear completely afterwards. If this noisy posterior probability exceeds the defined threshold, the corresponding event is deemed to be present in the audio clip. This noise can lead to false positives and increase the error rate. Therefore, sliding window method helps to smooth these singularities and provide a better detection performance [116].

8.2.3 Conclusions

In this section, the problem of rare sound event detection was addressed. A deep learning approach using a combination of CNNs and RNNs was applied to model the spatial and temporal properties inherent in sound events. The outputs from the CRNN were smoothed with *sliding ensemble* and then a fixed threshold was applied to obtain final

temporal probabilities and to detect the precise onset of the events. It can be concluded that the application of ensembles with different time steps leads to stronger predictions and removes unwanted noise from input audio signals. The CRNN approach has less complexity compared to an RNN-only system, as the applied 2D-CNN provides abstraction and reduces the overall number of trainable parameters. Moreover, the inclusion of LSTM cells leads to more accurate onset predictions, and the results showed that long term temporal properties are efficiently modelled as compared to the individual architectures (cf. CNN and Baseline FFNN in Table 8.7) [128]. As part of future work, it might be beneficial to experiment with more time step ensembles and larger amount of synthesised data, as these factors were shown to be very effective for the proposed task [116].

8.3 Audio-Based Game Genre Classification

Video games are an interactive audio-visual and tactile medium, with the soundscape playing a key role in the overall gaming experience [190]. The development of game audio can be considered as the result of a series of technological, economic, ideological, social, and cultural pressures [191]. Further, elements such as genre and audience expectations often shape gameplay audio [191].

Key audio events in games include: *vocalisations* of game characters, *sound effects* relating to gameplay, *ambient effects* relating to atmosphere, and the *music* of the game [192]. The mix of these events within a particular game depends on gameplay mechanics and is highly related to the genre [190]. For example, action and shooting games such as the Call of Duty™ series will contain loud, sudden events including punches and gunshots; sports games such as the FIFA football series tend to have ongoing commentary voice-overs; racing games such as the Forza™ series contains a substantial amount of car noises including heavy accelerations and screeching brakes; finally, classic arcade games such as Super Mario™ have *chiptune*-based (8-bit synthesised electronic music) accompanying sounds.

Given the links between the audio content of a video game and its genre [190, 192] and the growing influence of AI in game development [193, 194], this section explores different machine learning based acoustic detection paradigms for the task of *Video Game Genre Classification*. As well as being a challenging machine learning task, this work has many potential real-world applications, for example:

- Development of a remote and unobtrusive tool to automatically monitor game usage. Such a tool could allow parents to better track their child's video game habits monitoring total play-time and check if the game is age appropriate [195]. Any such tool should of course be developed under a clear ethical framework ensuring that it has the goal of monitoring strictly for health and wellness purposes, whilst maintaining and protecting privacy [196].

- General activity monitoring, e. g. in smart homes. An analogous example are apps that collect TV-viewing data for advertisers using a smartphone's microphone; according to a recent report in the New York Times[2], there are at least 250 such apps currently on the market.

- First step towards SHAZAM[3] for game audio which can, e. g. identify game genres, game's name, game walkthroughs, or game tutorials based on a short audio sample played and using the microphone on a device.

- As a simple objective aid for game designers to boost the decision making process for choosing the proper game sound of new games and to validate that the game has

[2] Retrieved November 06, 2018, from `https://nyti.ms/2E7Iins`

[3] `https://www.shazam.com/`

a suitable soundscape for a particular genre.

- Aiding the automatic generation of game genre-specific music tracks.

- Monitoring of games on YouTube and other social media platforms. As well as aiding the automatic segmentation of gameplay clips posted on social media into semantically meaningful chunks.

- Automatic social media-based game retrieval system.

In this section, a new video game corpus, *Game Genre by Audio + Multimodal Extracts* (G^2AME), collected by the author and his colleagues will be introduced [197]. G^2AME is a collection of 1 566 audio clips taken from 300 different video games and grouped into six genres.

The classification paradigms explored for the G^2AME corpus have been adopted from techniques successfully used in other acoustic-detection tasks. The baseline system is based on acoustic feature sets which are extracted from audio clips of different video games using the openSMILE toolkit [8]. These feature sets are primarily used in computational paralinguistics-based recognition tasks [198, 199]. However, they have also been used for movie genre classification [200], and music genre classification [201, 202]. The efficacy of these standard acoustic feature sets is compared with state-of-the-art Deep Spectrum features (cf. Chapter 4) [16] and their quantised representation BoDF [88]. A detailed description of these systems can be found in Chapter 4 and Section 4.2.3 [197].

Pre-trained CNNs have been chosen for the deep feature extraction from audio data for the following reasons: first, because of the richness of the time-frequency information in spectrograms, which are fed as input images into the pre-trained CNNs, local structures relating to properties, such as loudness, pitch, rhythm, and spectral energy distribution are inherently present, and are in turn readable for the CNNs. This is verified by the strong performance of the Deep Spectrum features in a range of audio tasks, e. g. [4,16,18,62, 136]. Second, fine-tuning or training a new deep learning model on the G^2AME dataset, in which data quantity is limited, may highly increase the risk of overfitting to the training data [197].

8.3.1 Data and Procedure

To evaluate the G^2AME corpus (cf. Section 8.3.1), experiments to predict 6 game genres have been performed, including conventional and state-of-the-art machine learning approaches. In total, three sets of features have been extracted from the dataset: i) handcrafted acoustic feature sets (cf. Section 8.3.1.2), ii) Deep Spectrum features (cf. Section 8.3.1.3), and iii) BoDF (cf. Section 8.3.1.4).

Table 8.8: The distribution of audio clips in the G^{2}AME corpus across the six genres. The number of 5 s clips per genre is denoted in parentheses.

Genre	**Videos** (5 s Clips)
ACS	258 (3096)
ARP	205 (2460)
FHT	330 (3960)
RCG	296 (3552)
SPT	266 (3192)
SWB	211 (2532)
$\sum$	1 566 (18792)

8.3.1.1 G^{2}AME Corpus

The Game Genre by Audio + Multimodal Extracts (G^{2}AME) dataset contains 1 566 unique gameplay videos of 300 individual games. Each recording is converted to 16 kHz wav files cut into chunks of one minute in length. The total net playtime of G^{2}AME is 26 hours of gameplay. Further, each clip is cut to 12 individual 5 second chunks which are later used as a basis for feature extraction and classification in the non-BoDF systems. This results in a total of 18 792 audio chunks (cf. Table 8.8).

Using the popular online shopping platform *Amazon*[4] as a guide, the games and according audio clips are categorised into six different genre groups:

(i) *Action or Shooter* (ACS) games; 258 instances picked from games such as Battlefield 1, Assassin's Creed, Dark Souls, Diablo, or Call of Duty.

(ii) *Arcade or Platform* (ARP) games; 205 instances picked from games such as Sonic the Hedgehog, Donkey Kong, Golden Axe, Pac-Man, or Super Mario Brothers.

(iii) *Fighting* (FHT) games; 330 instances picked from games such as Mortal Kombat, Street Fighter, or Tekken.

(iv) *Racing* (RCG) games; 296 instances picked from games such as Forza, Gran Turismo, or Need For Speed.

(v) *Sports* (SPT) games; 266 instances picked from games such as FIFA, NBA, MLB, Pro Evolution, or WWE2.

(vi) *Simulation or World Building* (SWB) games; 211 instances picked from games such as Age of Empires, Minecraft, Tropico, Warcraft, or The Sims.

[4] www.amazon.com

Two different versions of the same game are treated as one game, e. g. the football games FIFA 16 and FIFA 15 are both considered examples of the FIFA game. Each genre contains clips from 50 distinct video games. For the used *Cross Validation* (CV) scheme, each of the 10 folds contains instances of 5 distinct games from every genre. This provides a 'game independence', ensuring that the machine learning algorithms do not focus on recognising specific games instead of their respective genres. For the machine learning experiments in this work, it is focused strictly on an audio-based approach for the following reasons:

- Practical use in a game monitoring application; the genre can be recognised from the distance, such as in a personal assistant device, with no need to see or analyse the screen content. Furthermore, audio processing, in general, is often considered more lightweight than visual processing, and hence it is potentially better suited for real-time classification in embedded devices.
- Audio is an essential part of video games that helps to enhance the gaming experience. In this regard, music, which is not visible, plays a key role in establishing atmospheric difference between various game genres. Audio also provides instant feedback to the player's inputs such as shooting a gun. This is an important factor to get a better analysis of player's gaming behaviour.
- Classification of the game genres is potentially more reliable from the audio modality. For example, role playing games, such as Dark Souls, Witcher 3, or Fallout 4 are often visually diverse making it harder to infer the genre. Audio also gives cues about the visually invisible objects, monsters, animals, or persons in a gameplay (e. g. a person behind a wall, or a monster hidden in bushes). Obtaining such information can improve the performance of a game analysis toolkit.

8.3.1.2 Acoustic Feature Sets

The conventional, expert-designed acoustic feature sets are extracted, which are used for the INTERSPEECH 2009 Emotion Challenge (IS09) [198] and the INTERSPEECH 2010 Paralinguistic Challenge (IS10) [199], with 384 and 1 582 features, respectively. For full information on the extraction and formation of these feature sets, the interested reader is referred to corresponding references, as well as to [203].

8.3.1.3 DEEP SPECTRUM Features

Using the DEEP SPECTRUM system, deep features are extracted from visual representations of audio data, including spectrograms, mel-spectrograms, chromagrams, and their temporal transitions (deltas). These audio plots have shown to be highly effective for various audio-based classification tasks [65, 204, 205]. The complete process of extracting DEEP SPECTRUM features is introduced in Chapter 4. For the sake of reproducibility, following configurations are provided to generate the deep representations.

8.3.1.3.1 Audio Plots All audio plots are computed from Hanning windows of width $w = 16$ ms and overlap of 8 ms samples. The mel-spectrograms are calculated from the log-magnitude spectrum by dimensionality reduction using 128 mel-filterbanks equally spaced on the mel-scale. The chromagrams are extracted using the default implementation provided by the *librosa* Python library [206]. Video games are often accompanied by scores or soundtracks, hence, chromagrams might be able to capture some genre specific musical characteristics. Furthermore, the first order derivatives (deltas) of the mel-spectrograms and chromagrams are computed. For more information about the audio plots, the interested reader is referred to Section 4.2.1.

To highlight the audio similarities and differences that potentially exist between the game genres, the visual representations of audio samples contained in the six different classes are depicted in Figure 8.4. These samples are taken from an exemplar game within each genre.

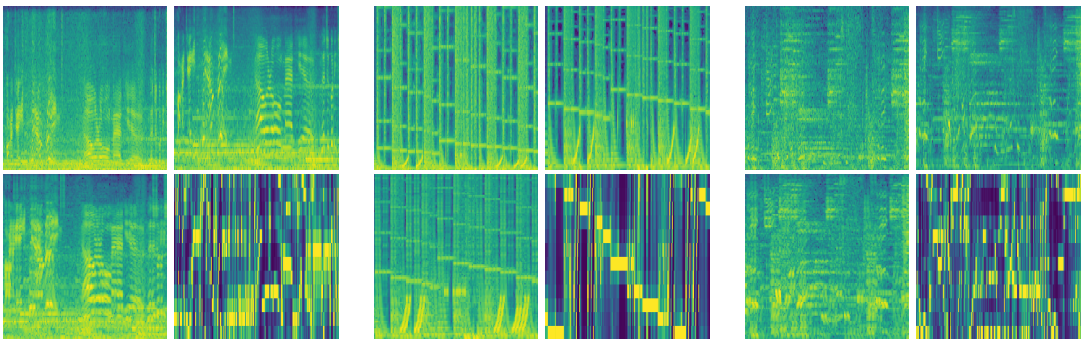
For the spectrogram plots, three different colour mappings are also used: *viridis*, *hot*, and *Vega20b*. It is during testing (cf. Section 8.3.2) that the optimal colour map for the spectral and chroma features is identified.

8.3.1.3.2 CNN-Descriptors To form suitable feature representations from the plots described before, the activations of the 4 096 neurons on the second fully connected layer (*fc7*) of AlexNet [28] and VGG16 [30], and the activations of the 1 024 neurons on the last pooling layer of GoogLeNet [144] are extracted as feature vectors, because of their robustness for audio classification tasks [4, 16, 62].

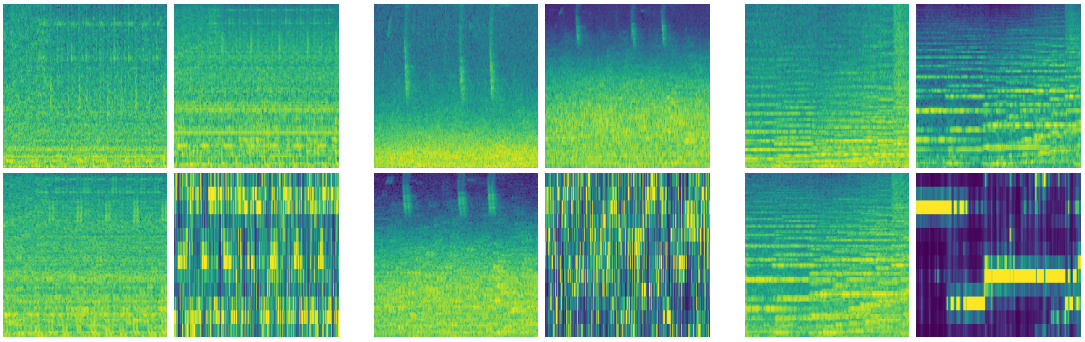
8.3.1.4 Bag-of-Deep-Features

The extracted DEEP SPECTRUM representations are quantised using OPENXBOW [145] to form the BODF. In order to achieve this, fixed length histogram representations of each audio recording are generated. As described in Section 4.2.3, this is done by identifying a set of 'deep audio words' from some given training data, and then quantising the original feature space, with respect to the generated codebook, to form the histogram representation. The histogram shows the frequency of each identified deep audio word in a given audio instance [145–147]. The features are then normalised to $[0, 1]$ and a codebook with fixed size from the training partition is random sampled. Afterwards, each input feature vector (from training and evaluation partitions) is assigned to a fixed number of its closest vectors from the codebook. Finally, the logarithmic term-frequency weighting to the generated histograms are used.

For the experiments, the codebook size (*cs*) and the number of assigned codebook words (*cw*) are optimised with $cs \in \{100, 200, 500, 1\,000\}$, $cw \in \{1, 10, 20, 50, 100, 200\}$.



(a) Alice: Madn. Ret. (ACS) (b) Dr. Robotnik (ARP) (c) AquaPazza (FHT)



(d) Daytona USA (RCG) (e) Cricket (SPT) (f) Cossacks (SWB)

Figure 8.4: Example spectrograms, mel-spectrograms, delta-mel-spectrograms and chromagrams (left-to-right and top-to-bottom) of audio samples contained in the six different classes of the G^{2}AME corpus.

8.3.2 Results

In order to predict the class labels for the audio instances in the G^{2}AME corpus, the feature sets are evaluated using 10-fold CV with linear SVM classifier. The evaluation measure is *Unweighted Average Recall* (UAR) with the 6-class chance level being 16.7 % UAR for all experiments. The open-source linear SVM implementation provided in the scikit-learn machine learning library is used [24]. Feature standardisation is applied to the conventional acoustic feature sets. For the DEEP SPECTRUM features, both standardisation and normalisation have been found to negatively impact classifier's performance. The built-in balancing option of the SVM classifier is used to counteract the slight imbalance of the dataset. Using the 10 fold CV setup, the classifier's complexity parameter is optimised in 10 steps, equally spaced on a logarithmic scale between 10^{-9} and 10^{0}.

An extensive series of experiments has been conducted to evaluate the performance of the extracted feature sets using the proposed classifiers.

First, the performance of the acoustic feature sets is evaluated in Section 8.3.2.1. The classification results are then obtained for the DEEP SPECTRUM features (cf. Sec-

Table 8.9: Performance of the SVM classifier using different openSMILE audio functionals on the G^{2}AME corpus. UAR on concatenated folds is provided. Mean and standard deviation are reported in parentheses.

Acoustic feature set	C	UAR [%]
IS09	10^{-6}	49.6 (49.2 $\pm$ 4.1 %)
IS10	10^{-6}	55.2 (54.7 $\pm$ 4.5 %)

tion 8.3.2.2). Afterwards, the effect of quantising the best performing Deep Spectrum feature sets for each CNN-descriptor is evaluated (cf. Section 8.3.2.3). In Section 8.3.2.4, model (late) fusion is performed on the Deep Spectrum and acoustic feature sets. Finally, the best classification result is compared with human performance (cf. Section 8.3.2.5). Moreover, statistical T-tests are performed, comparing the results from acoustic feature sets with the best Deep Spectrum results and the overall best performance in a pairwise fashion to determine if they are statistically different. The null-hypothesis is rejected at a significance level of $p < 0.05$. For each comparison, the p-values can be found in Table 8.13. The results of the CV are checked for normality using a Shapiro-Wilk test [207, 208].

8.3.2.1 Acoustic Feature Sets

It can be observed that the larger feature set (IS10) with 1 582 features outperforms the smaller one (IS09) with 384 features (cf. Table 8.9) reaching a maximum UAR of 55.2 %. It is shown that there is statistically significant difference between these two feature sets (cf. Table 8.13).

8.3.2.2 Deep Spectrum Features

The comparison of different feature plots as a basis for Deep Spectrum extraction (cf. Table 8.11) indicates that chromagrams – in contrast to their relevance to music analysis – do not provide suitable input for ImageNet pre-trained CNNs. This is partially explained by the inherent "unnatural" look of these plots and the lack of local dependencies between the pixels, leading to an inability to extract informative structural features using the CNN models. Conversely, applying mel-spectrograms slightly improves performance for both AlexNet and GoogLeNet. Using the deltas, however, the performance drops for all CNNs. The overall best performance is achieved by AlexNet features extracted from mel-spectrograms with a UAR of 59.9 %.

Table 8.10: Performance of the SVM classifier using DEEP SPECTRUM features extracted from spectrogram plots with different colour maps by three CNN-descriptors on the G^{2}AME corpus. UAR on concatenated folds is provided. Mean and standard deviation are reported in parentheses.

UAR [%]	AlexNet	GoogLeNet	VGG16
hot	58.1 (57.5 $\pm$ 3.2 %)	55.3 (54.7 $\pm$ 4.6 %)	58.8 (57.9 $\pm$ 4.0 %)
Vega20b	55.2 (54.7 $\pm$ 3.8 %)	52.3 (51.5 $\pm$ 4.1 %)	55.7 (54.8 $\pm$ 4.2 %)
viridis	58.6 (57.9 $\pm$ 3.7 %)	54.7 (54.0 $\pm$ 3.8 %)	58.3 (57.6 $\pm$ 3.1 %)

Table 8.11: Performance of the SVM classifier using best performing DEEP SPECTRUM features on the G^{2}AME corpus from Table 8.10. UAR on concatenated folds is provided. Mean and standard deviation are reported in parentheses.

UAR [%]	AlexNet (viridis)	GoogLeNet (hot)	VGG16 (hot)
spectrograms	58.6 (57.9 $\pm$ 3.7 %)	55.3 (54.7 $\pm$ 4.6 %)	58.8 (57.9 $\pm$ 4.0 %)
mel-spectrograms	59.9 (59.2 $\pm$ 4.7 %)	58.3 (57.6 $\pm$ 4.1 %)	58.7 (57.8 $\pm$ 4.2 %)
$\vec{\Delta}$mel-spectrograms	56.5 (55.7 $\pm$ 4.9 %)	54.8 (53.9 $\pm$ 4.5 %)	55.2 (54.3 $\pm$ 4.6 %)
chroma	46.2 (45.4 $\pm$ 4.1 %)	46.2 (45.3 $\pm$ 3.1 %)	47.6 (46.6 $\pm$ 2.9 %)
$\vec{\Delta}$chroma	41.1 (40.3 $\pm$ 3.6 %)	38.7 (37.9 $\pm$ 2.4 %)	41.1 (40.3 $\pm$ 2.9 %)

8.3.2.3 Bag-of-Deep-Features

For the BODF representations, the best performing feature plots for each CNN are chosen, i. e. mel-spectrograms for both AlexNet and GoogLeNet, and regular spectrograms for VGG16. The BODF parameters codebook size *cs* and number of assigned codebook words *cw* as well as the SVM's complexity parameter *C*. The results in Table 8.12 show a maximum UAR of 66.9 % achieved by BODF with $cs = 500$ and $cw = 25$ formed from DEEP SPECTRUM features and GoogLeNet mel-spectrograms as CNN-descriptor. It is also shown that there is statistically significant difference between GoogLeNet BODF system and the other tested approaches (cf. Table 8.13). A confusion matrix for this system is displayed in Figure 8.5b.

8.3.2.4 Fusion Experiments

In addition to the quantisation method employed in Section 8.3.1.4, late fusion is also performed on the different chunk-level feature representations. The best performing DEEP SPECTRUM features are fused with each of the three audio functional feature sets and also all four of them are concatenated together. Note that an early fusion was also

Table 8.12: Performance of the SVM classifier using BoDF representations of the best Deep Spectrum features on the G^{2}AME corpus. *cs*: codebook size; *cw*: number of assigned codebook words; *C*: SVM classifier's complexity. UAR on concatenated folds is provided. Mean and standard deviation are reported in parentheses. The best classification result is highlighted with a light grey shading.

UAR [%]	*cs*	*cw*	*C*	10-fold CV
AlexNet mel-spectrograms (viridis)	1 000	200	10^{-6}	66.7 (66.0 ± 4.3 %)
	1 000	100	10^{-5}	66.6 (64.7 ± 4.7 %)
GoogLeNet mel-spectrograms (hot)	500	25	10^{-5}	**66.9** (66.3 ± 4.0 %)
	500	100	10^{-6}	66.3 (64.8 ± 4.2 %)
VGG16 spectrograms (hot)	1 000	100	10^{-6}	65.1 (64.6 ± 6.4 %)
	1 000	50	10^{-6}	64.8 (64.2 ± 5.5 %)

Table 8.13: *p*-values for T-test scores comparing 10-fold CV results of different configurations. Except for AlexNet vs. VGG16 and VGG16 vs. IS10 the difference between other feature sets is statistically significant.

	GoogLeNet BoDF (66.9)	AlexNet (59.9)	VGG16 (58.8)	IS09 (49.6)	IS10 (55.2)
GoogLeNet BoDF (66.9)	1.0	$4 \cdot 10^{-4}$	$4 \cdot 10^{-4}$	$5 \cdot 10^{-8}$	$2 \cdot 10^{-5}$
AlexNet (59.9)	-	1.0	0.522	$1 \cdot 10^{-4}$	$4.9 \cdot 10^{-2}$
VGG16 (58.8)	-	-	1.0	$2 \cdot 10^{-4}$	0.128
IS09 (49.6)	-	-	-	1.0	0.014
IS10 (55.2)	-	-	-	-	1.0

attempted, however, initial analysis revealed that this approach was not suitable.

The late fusion scheme uses the trained and optimised SVM models obtained in Sections 8.3.2.1 and 8.3.2.2 and combines their predictions by majority vote. These results (cf. Table 8.14) further indicate that the different feature sets are not complementary.

8.3.2.5 Comparison with Human Performance

To gain perspective into how well the best classification approach performed (cf. Table 8.12), human classification tests have been conducted through the browser-based crowdsourcing platform iHEARu-PLAY[5] [209].

For the perception task, the human raters were presented with one 5 s clip (picked at random) for each file. A total of 12 individuals (7 male, 5 female, average age 27.3, non-professional gamers) completed the full classification task on the iHEARu-PLAY platform. The average per rater decision time was 3.31 s. The best human performance

[5] `https://www.ihearu-play.eu`

Table 8.14: Performance of late fusion using a linear SVM classifier on the G^{2}AME corpus. A majority vote is employed using the best individual models obtained during previous experiments. UAR on concatenated folds is provided. Mean and standard deviation are reported in parentheses.

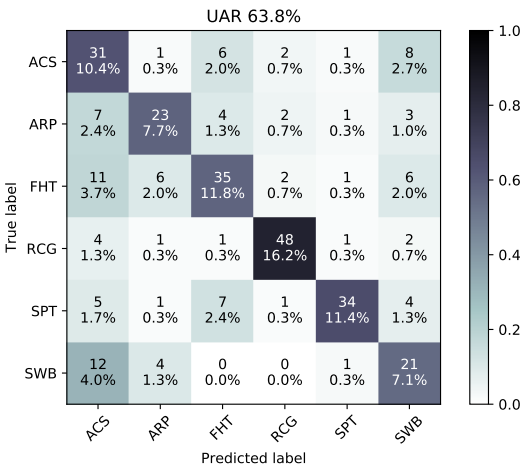
Late fusion	UAR [%] 10-fold CV
DEEP SPECTRUM + IS09	56.0 (55.4 $\pm$ 4.0 %)
DEEP SPECTRUM + IS10	58.3 (57.7 $\pm$ 4.2 %)
all features combined	57.8 (57.4 $\pm$ 4.5 %)

was 63.8 % UAR, which is 3.1 percentage points less than the best classification result. The confusion matrix for this prediction is given in Figure 8.5a.

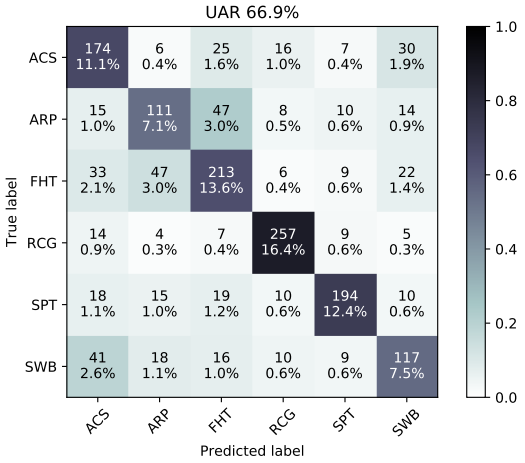
Whilst the raters' scores gives a reference baseline on which to gauge the machine learning approaches, it should be noted that due to the differing amounts of training, as well as the CV paradigm used in the machine learning approaches, such a comparison should not be considered like-for-like. In addition, the models were trained on a large amount of data (cf. Table 8.8) which the raters did not have access to. However, given the prevalence of video games in today's society [192, 210], one cannot say that the raters received no training. Further, the terminology associated with each genre label would have helped the raters form preconceptions of what a 'typical' audio clip from a particular genre should sound like. Despite these factors, the stronger performance of the deep learning approach indicates the suitability of using machine learning for the task of audio-based video game genre classification.

8.3.3 Conclusions

This section explored machine learning paradigms for the task of audio-based video genre classification. Potential real-world applications of such a system were listed. In Section 8.3.1.1, the novel G^{2}AME dataset was presented, which includes 1 566 unique gameplay samples taken from 300 individual video games. In Sections 8.3.2.2 and 8.3.2.3, the efficacy of the deep learning paradigms proposed by the author and his colleagues have been analysed and compared with the performance of the baseline system in Section 8.3.2.1. The results in Section 8.3.2 indicated that BODF, a combination of DEEP SPECTRUM features and *Bag-of-Audio-Words* (BOAW), are well suited to the task of audio-based game-genre classification. This system achieved the strongest UAR of 66.9 % an improvement of 3.1 percentage points over humans performing the same task. Statistical T-tests were also performed to compare the results obtained from various feature sets in a pairwise manner. It was demonstrated that there are statistically significant differences between the overall best result and the results obtained from other proposed



(a) *Confusion Matrix from classification labels for 297 audio files gained by the best human rater.*



(b) *Confusion Matrix from classification labels for the 1 566 audio files in the G²AME dataset gained by the best* BoDF *system with* $cs = 500$ *and* $cw = 25$.

Figure 8.5: Confusion matrices from the human performance and our best classification result.

feature vectors (cf. Table 8.13).

An in-depth analysis of the results reveals that *Racing* games, which normally feature a substantial amount of automotive noises, were the easiest to recognise. *Simulation and World Building* games, which showed a high confusion with *Action or Shooter* genre, were the most difficult to analyse.

For the future work, it is of great interest to grow the G²AME dataset. The visual-based, linguistic-based (YouTube comments), and multimodal classification should also be considered. Given the complexity of video game labels, other future work should include advanced clustering techniques, whilst associating each video game with a set of genre labels, rather than applying the current method of assigning each a single label in a multimodal framework.

9

In-the-Wild Speech, Vocalisations, and Sentiment

The growing amount of multimedia material publicly available online, including produced content and non-acted personal home videos, represents an untapped wealth of data for research purposes. For example, Gemmeke et al. introduced *AudioSet*, which is a large-scale dataset of 2.1 million human-labelled 10-second audio clips obtained from YouTube videos [93]. Perception and classification of such in-the-wild audio recordings is a particular challenging aspect of computer audition. These audio samples typically contain a variety of confounding effects, such as non-stationary noise and less than ideal microphone placements. In this regard, this chapter analyses the suitability and the performance of the novel *Bag-of-Deep-Features* (BoDF) for classification of in-the-wild human vocalisation and speech types (cf. Section 9.1) [88], and sentiments (cf. Section 9.2) [62]. The datasets on which the experiments are conducted have also been sourced from YouTube [88, 151, 211].

9.1 In-the-wild Speech and Vocalisation

In order to minimise the time and computational cost for processing high-dimensional features, and reduce the amount of noisy representations extracted from real-world audio data, it is beneficial to compress the feature space [145]. In this regard, this section investigates the applicability and the efficacy of utilising BoDF representations (cf. Section 4.2.3) for various in-the-wild speech and vocalisations datasets. Furthermore, the results obtained from BoDF representations will be compared with non-quantised Deep Spectrum features, and two baseline systems using MFCCs and conventional acoustic feature sets [8, 151, 198].

9.1.1 Data and Procedure

First, mel-spectrograms are created from chunked audio recordings in the corpus (cf. Section 4.2.1). They are then forwarded through various image classification CNNs (cf. Section 4.2.2) to extract the DEEP SPECTRUM features (cf. Chapter 4). Afterwards, BODF are created by quantising the DEEP SPECTRUM features (cf. Section 4.2.3). In the final step, a classifier has been trained on both DEEP SPECTRUM features and BODF to analyse the efficacy of the generated features.

9.1.1.1 Dataset

The corpus introduced in [151], which has been used for the experiments in this section, includes the following six real-world audio databases containing different human speech and vocalisation types:

1. *Freezing:* 785 recordings picked from videos in which the speech is produced by an individual shivering with cold.

2. *Intoxication:* 1 221 language independent recordings picked from videos in which the speech is produced under the influence of drugs.

3. *Screaming:* 564 recordings picked from videos in which people are screaming when they are scared.

4. *Threatening:* 1 093 language independent recordings picked from videos in which the speech is perceived by the annotators to be of a threatening manner.

5. *Coughing:* 3 659 recordings picked from videos in which people are coughing during a conversation or a talk.

6. *Sneezing:* 529 recordings picked from videos in which people are sneezing during a conversation or a talk.

These datasets are based on the concept of acoustic surveillance [2]. The first four topics are related to audio-based surveillance for security purposes in noisy public places. The latter two topics, related to the monitoring of everyday activity – in terms of, e. g. personal health – in common, relatively quiet environments, such as home or office [2]. All corpora offer a two-class classification problem, i. e. they have a target class, e. g. *freezing* or *intoxication* and a 'normal speech' class which contains audio samples that are not affected by the target class. All audio data is mono at a rate of 44.1 kHz. For full details on the construction of theses datasets the interested reader is referred to [151]. The specifications of each database is given in the Table 9.1.

Table 9.1: Specifications of each database. l_{total}: the total length of the data set; l_{min} and l_{max}: the minimum and maximum lengths of the audio recording; SD: standard deviation; n: the number of all audio recordings in each set. s : the number of 0.5 s segments, i. e. the number of frames of input mel-spectrograms, denoted in parentheses. c_{ratio}: the class ratio for each data set (target class:'normal speech').

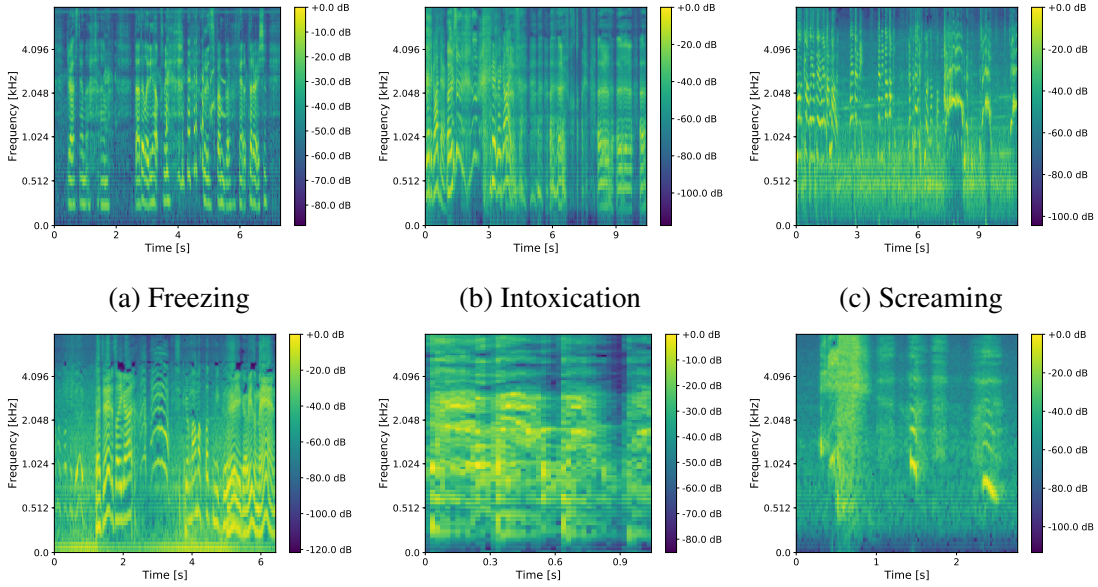
	Train					Evaluation				
Tasks	l_{total}	l_{min}/l_{max}	SD	n (s)	c_{ratio}	l_{total}	l_{min}/l_{max}	SD	n (s)	c_{ratio}
Freezing	75.9 m	2.0 s/29.4 s	5.8 s	614 (8813)	2 : 1	22.4 m	2.0 s/28.6 s	5.9 s	171 (2595)	1.1 : 1
Intoxication	139.7 m	2.0 s/29.9 s	6.5 s	1069 (16200)	0.9 : 1	16.7 m	2.0 s/24.8 s	5.3 s	152 (1930)	1.8 : 1
Screaming	53.6 m	2.0 s/29.9 s	7.6 s	375 (6192)	1.2 : 1	22.0 m	2.1 s/29.9 s	5.5 s	189 (2505)	1.4 : 1
Threatening	106.6 m	2.0 s/29.8 s	7.4 s	652 (12360)	6 : 1	45.8 m	2.0 s/29.2 s	5.2 s	441 (5271)	0.6 : 1
Coughing	94.3 m	0.5 s/28.8 s	3.5 s	2088 (10336)	2.9 : 1	63.9 m	0.5 s/23.2 s	2.7 s	1571 (6935)	2.2 : 1
Sneezing	6.7 m	0.5 s/8.0 s	1.3 s	238 (691)	0.9 : 1	9.2 m	0.5 s/9.3 s	1.4 s	291 (967)	1 : 1
Σ	476.8 m	–	–	5036 (54592)	– –	180 m	–	–	2815 (19933)	– –

9.1.1.2 Deep Spectrum Features

The DEEP SPECTRUM features are extracted in three stages (cf. Chapter 4). First, as described in Section 4.2.1, mel-spectrograms with a window width of $w = 0.5$ s and an overlap of 0.25 s are extracted from the chunked audio contents. They are then forwarded through four different architectures of pre-trained CNNs, including AlexNet [28], VGG16 and VGG19 [30], and GoogLeNet [144]. In the third and final step, the activations of AlexNet's seventh layer (*fc7*), VGG16's and VGG19's second fully connected layer, and the activations of the last pooling layer of GoogLeNet are extracted as large feature vector for each input mel-spectrogram. In Figure 9.1, the audio similarities and differences that potentially exist between different classes in the corpora are highlighted by showing an example mel-spectrogram from each target class.

9.1.1.3 Bag-of-Deep-Features

The same procedure as introduced in Section 8.3.1.4 is applied for generating BODF features [88]. For the experiments, the size of the codebook (cs) and the number of assigned codebook words (cw) are optimised with $cs \in \{10, 20, 50, 100, 200, 500, 1\,000\}$, $cw \in \{1, 10, 25, 50, 100, 200, 500\}$ and evaluated on the evaluation partition using a linear SVM classifier. For this purpose, the classifier's complexity parameter (C) is optimised on a logarithmic scale between 10^{-9} and 10^{0} with a factor of 10. The best performing codebook is then applied for evaluation on the test set.



(a) Freezing (b) Intoxication (c) Screaming

(d) Threatening (e) Coughing (f) Sneezing

Figure 9.1: Example mel-spectrograms extracted from the target classes contained in the six different corpora. Relatively high f_0 of the *Threatening* class, wide band spectra for *Coughing* and *Sneezing* classes,and narrow band spectra for the *Freezing*, *Intoxication*, and *Screaming* classes, can be observed.

9.1.1.4 Baseline Features

The DEEP SPECTRUM and BODF features are compared with two strong baseline feature sets introduced in [151]: (i) the INTERSPEECH 2009 Emotion Challenge feature set (IS09) with a total number of 384 features, and (ii) a 39-dimensional MFCC feature set.

9.1.1.4.1 The INTERSPEECH 2009 Emotion Challenge feature set (IS09) This feature set contains 384 features [198]. The total number of attributes is obtained by multiplying 16 *Low-Level Descriptors* (LLDs) $\times$ 2 (as the delta coefficients of the LLDs are also included) $\times$ 12 statistical functionals. In detail, the 16 LLDs in *IS09* set are: *Root Mean Square* (RMS) frame energy, pitch frequency (normalised to 500 Hz), *Zero-Crossing-Rate* (ZCR) from the time signal, *Harmonics-to-Noise Ratio* (HNR) by autocorrelation function, and 12 MFCCs [198]. To each LLD, the delta coefficients are additionally computed. Afterwards, 12 functionals, including mean, standard deviation, minimum and maximum value, kurtosis, skewness, relative position, and range as well as two linear regression coefficients with their MSE are applied on a chunk basis. Therefore, the each feature vector per chunk contains $16 \times 2 \times 12 = 384$ attributes [198].

9.1.1.4.2 Mel-Frequency Cepstral Coefficients Using OPENSMILE [8, 9], a 39-dimensional MFCC representation is extracted from the audio signals with a Hanning window of width $w = 25$ ms and overlap 10 ms. Subsequently, 12 MFCCs from 26 mel-frequency bands obtained from the FFT power spectrum are computed. Afterwards, a cepstral liftering filter with a weight parameter of 22 is applied. Finally, 12 delta, 12 acceleration coefficients, and 3 logarithmic energies are appended to the MFCC. The frequency range of the mel-spectrum is set from 0 to 8 kHz.

9.1.2 Results

First, the classification results for the non-quantised DEEP SPECTRUM features are obtained. The robustness of quantising the representations (BODF) for all four CNN-descriptors is then evaluated (cf. Section 9.1.2.2). Finally, early (feature) and late (model) fusion experiments for various combinations of the CNN-descriptors are performed (cf. Section 9.1.2.3).

9.1.2.1 Classifier and Evaluation Metric

To predict the class labels, a linear SVM classifier is trained. The evaluation metric is UAR, as this measure gives equal weight to all classes. For the classifier, the open-source linear SVM implementation provided in the *scikit-learn* machine learning library [24] is used. For the DEEP SPECTRUM features, standardisation and normalisation are not applied, as they have been found to negatively impact classifier performance. Moreover, a SVM is preferred over DNN as a classifier for two reasons: first, BODF is a sparse feature representation and SVMs are effective at handling sparse data [212, 213], and second, the datasets are too small to train a DNN.

9.1.2.2 Performance of the Representations

For the non-quantised DEEP SPECTRUM (nqDS) features, majority voting is applied to obtain the prediction for a complete audio recording from its chunk-level results. It is shown that the results – despite being strong – are behind the best baseline feature and almost all BODF results (cf. Table 9.2) [88].

The results in Table 9.2 show that quantisation improves the results for all CNN-descriptors. The results also demonstrate the strength of the BODF outperforming the best baseline results for the acoustic surveillance databases *Freezing*, *Intoxication*, *Threatening*, and *Screaming*. It is observed that BODF worked best on longer audio chunks, as opposed to shorter ones from the *Coughing* and *Sneezing* datasets. It is assumed that this effect is due to the lack of discriminating information in the shorter chunk spectrograms leading to weaker DEEP SPECTRUM representations. It is worth noting that for both *Coughing* and *Sneezing*, BODF consistently outperforms DEEP SPECTRUM features

Table 9.2: Classification results of each paralinguistic task from the baseline paper [151] by the functionals (Func.) of the IS09 feature set, and BoAW representations, compared with the results from the non-quantised Deep Spectrum features (nqDS) and BoDF representations. For the non-quantised Deep Spectrum features, majority voting is used to obtain the prediction for a whole audio recording from its chunk-level results. The chance level for each task is 50.0 % UAR. The best result for each corpus is highlighted with a light grey shading.

UAR [%]	IS09 [151]		MFCCs [151]	AlexNet		VGG16		VGG19		GoogLeNet	
	Func.	BoAW	BoAW	nqDS	BoDF	nqDS	BoDF	nqDS	BoDF	nqDS	BoDF
Freezing	70.2	67.5	65.6	62.5	70.4	71.3	**72.9**	67.9	69.1	67.3	71.6
Intoxication	64.7	72.6	66.7	60.3	61.9	58.2	64.7	55.4	71.3	63.1	**73.6**
Screaming	89.2	97.0	94.0	94.9	**98.5**	96.8	96.7	94.7	98.2	89.8	94.3
Threatening	73.8	66.3	67.0	72.2	76.4	70.7	73.9	70.6	70.3	70.5	**77.3**
Coughing	95.4	96.7	**97.6**	94.3	95.3	94.5	95.3	94.2	95.2	91.0	92.0
Sneezing	79.3	76.4	**79.8**	74.0	74.6	71.8	74.9	76.8	79.4	64.0	71.8

adding evidence to the assumption that quantisation improving system robustness, while compressing the feature space [88].

9.1.2.3 Early and Late Fusion

Both early and late fusion schemes are applied to the features and the models obtained from the proposed BoDF, in order to investigate their complementarity [88]. For early (feature-level) fusion, the Deep Spectrum features extracted from the chunked (0.5 s) audio recordings using the mentioned CNN-architectures are combined. Afterwards, BoDF representations of those features are built analogous to the non-fusion systems outlined in Section 9.1.1.3. A linear SVM is used for the classification task, and its cost parameter is optimised on a logarithmic scale between 10^{-9} and 10^{0} with a factor of 10. The late fusion scheme combines the predictions of the best BoDF models for each dataset obtained in previous experiments in a majority vote.

The results achieved by different configurations of these two fusion schemes on all six databases are displayed in Table 9.3. For *Sneezing* and *Coughing*, results are slightly improved over non-fusion systems but still do not reach the baseline performance in Table 9.2. In addition, small performance boosts of the early fusion models over non-fusion systems was observed for the *Intoxication* and *Screaming* datasets. Moreover, a larger increase in UAR was noticed for the *Freezing* dataset using a late fusion system of BoDF models based on the Deep Spectrum features obtained from AlexNet, VGG16, and GoogLeNet. However, as there is no consistent pattern, it is difficult to interpret the differences between the early and late fusion results of the CNN models. It is shown that the early fusion of all features for all tasks, except for *Screaming*, leads to stronger

Table 9.3: Performance of early and late fusion strategies for the CNN-descriptors using linear SVM classifiers on the databases. UAR is used as the measure. For early fusion, the linear SVM classifier's complexity parameter is optimised on a logarithmic scale between 10^{-9} and 10^{0} with a step size of 10. For late fusion, a majority vote is employed on the test set using the best individual models obtained during previous experiments. AlexNet is denoted as A., VGG16 as V16, VGG19 as V19, and GoogLeNet as G. The fusion results for each corpus which are better than the results given in Table 9.2 are highlighted with a light grey shading. The chance level for each task is 50.0 % UAR.

UAR [%]	Early fusion						Late fusion				
	A.+V16	A.+V19	A.+G.	V16+G.	V19+G.	All	A.+V16+V19	A.+V16+G.	A.+V19+G.	V16+V19+G.	All
Freezing	69.2	71.3	68.4	74.1	70.4	71.2	70.4	**76.3**	70.4	71.2	68.5
Intoxication	65.7	73.0	67.1	68.4	67.8	**73.8**	68.3	67.7	68.8	63.9	60.9
Screaming	98.5	**99.1**	97.8	97.1	**99.1**	98.2	98.0	98.0	98.2	98.2	98.9
Threatening	74.8	72.8	76.0	73.2	72.3	73.1	73.1	76.3	75.2	72.2	68.9
Coughing	96.0	95.5	95.4	95.6	95.3	96.5	96.3	96.3	95.2	95.2	95.3
Sneezing	76.3	77.7	76.3	75.6	78.0	79.8	77.0	77.7	79.1	79.8	74.1

performance than combining all models by late fusion. Based on these findings, it can be assumed that fusing high-level CNN features can lead to stronger performance than fusing the predictions of the trained models.

9.1.3 Conclusions

Despite representation learning with DNNs having shown superior performance over expert-designed feature sets in a range of machine learning recognition tasks, such approaches have not been widely explored within the domain of noisy, in-the-wild audio classification. In this regard, the results (cf. Tables 9.2 and 9.3) indicate that state-of-the-art image classification CNNs are capable of providing strong feature sets on real-world audio recognition [88]. Further, the strength of bagging the DEEP SPECTRUM features as a means of reducing the noise present in the feature space was demonstrated. It was shown that using BODF, it is possible to improve upon almost all results obtained from the non-quantised DEEP SPECTRUM features, whilst compressing the feature space and increasing the real-time capability of the applied machine learning system. The results give strong evidence that the quantising step, when bagging deep features, can be considered as a quasi-filtering process which, in general, improves system robustness. Finally, it was shown (cf. Table 9.3) that both early and late fusion can still increase the BODF classification results. These findings imply that features and models obtained from the applied CNN-descriptors are in most cases complementary.

In the future work, the efficacy of using BODF with other deep architectures, such as ResNet [46], Inception-v3 [144], and Inception-v4 [214] should be considered. It is highly recommended to explore the benefits of fine-tuning the pre-trained networks on larger in-

the-wild databases like *AudioSet* [93], *YouTube-8M* [215], or datasets for Acoustic Scene Classification and Sound Event Detection challenges [1, 178, 216].

9.2 Audio-Based Sentiment Analysis

Expressing emotions and sentiment is a central part of human communication [217]. We do this by laughing and smiling out of happiness, yawning out of boredom, or crying out of sadness. Some of those actions are performed quietly, so one can only recognise them via visual information, but other actions such as laughing or crying are audible [218].

In this section, an audio-based sentiment analysis is performed on movie reviews posted to YouTube [211]. This is essentially a polarity classification task [219], in which videos are assigned to a positive or negative class based on whether or not the presenter liked or disliked the movie they are reviewing. This task is generally achieved in a multimodal framework, which combines linguistic, speech, and visual cues. However, given the promising results recently published for speech-based emotion recognition using either BOAW [148] or DEEP SPECTRUM features [4], this section discusses the suitability of these representations for sentiment recognition [62].

9.2.1 Data and Procedure

The experiments for the sentiment analysis are performed on the audio modality of the *Movie Review Dataset* introduced by Wöllmer et al. [211]. First, the DEEP SPECTRUM features and MFCCs are extracted from the input audio files, as described in Section 9.1.1. Afterwards, BODF and *Bag-of-MFCCs* representations are created from each feature set using OPENXBOW [145]. Finally, the quantised feature sets are classified and the results are analysed.

9.2.1.1 Movie Review Dataset

The *Movie Review Dataset*, collected by Wöllmer et al. [211], consists of 359 YouTube clips in which people express their opinion on a selection of movies they have previously watched. The author and his colleagues used only the audio data from this dataset for the sentiment analysis [62]. The sentiment of the speaker in each recording is expressed as integer annotations on a *1–6* Likert scale, with *1* the negative and *6* the positive end of the sentiment spectrum. Based on this, the clips are separated into positive and negative, with average scores above *3* assigned to *positive* sentiment. The dataset is divided into three partitions: train, development (Devel.), and test. The statistics about each partition of the dataset is given in Table 9.4.

9.2.1.2 Bag-of-Deep-Features

Power spectrograms with Hanning windows of width $w = 16$ ms and overlap 8 ms are extracted using Python's matplotlib [220]. These spectrogram plots are then forwarded through AlexNet and the activations of its penultimate fully connected layer (*fc7*) with

Table 9.4: Distribution, average length and SD of all recordings from the Movie Review Dataset between train, development (Devel.), and test sets.

Statistics	Train	Devel.	Test	∑
# Videos	215	72	72	359
# Positive videos	125	42	42	209
# Negative videos	90	30	30	150
Average length (m:s)	2:32	2:27	2:30	2:31
SD (m:s)	0:40	0:41	0:44	0:41

4 096 neurons are obtained as feature vectors. These representations are then quantised using OPENXBOW to form the BODF. The same procedure as introduced in Section 8.3.1.4 is applied for creating BODF representations [88]. The codebook size (*cs*) and the number of assigned codebook words (*cw*) are optimised with $cs \in \{500, 1\,000, 2\,000, 4\,000\}$, $cw \in \{1, 10, 20, 50\}$.

9.2.1.3 Bag-of-MFCCs

The feature extraction toolkit OPENSMILE [8] is used to extract a 39-dimensional MFCC representation from the input audio signals with a Hanning window of width $w = 25$ ms and overlap 10 ms. A detailed account on the MFCC features is given in Section 9.1.1.4. Afterwards, the features are quantised using OPENXBOW. Finally, the codebook size (*cs*) and the number of assigned codebook words (*cw*) are optimised with $cs \in \{2\,000, 4\,000, 8\,000\}$, $cw \in \{1, 10, 20, 50\}$.

9.2.2 Results

The constructed BODF (cf. Section 9.2.1.2) and Bag-of MFCCs (cf. Section 9.2.1.3) are used for training a linear SVM, and input normalisation is applied. The imbalance of the datasets is counteracted by adjusting the weights for the two classes accordingly, i. e. 1.4 for negative and 1.0 for positive instances. Training and evaluating that system is performed using Weka's [221] LibLINEAR wrapper with the *L2-regularised L2-loss* solver [212]. Classifier's complexity parameter (*C*) is optimised on a logarithmic scale between 10^{-4} and 10^{0} with a factor of 10. For the best configurations on the development set, the features are then evaluated on the test set. All results for various configurations are given in Table 9.5.

The results in Table 9.5 indicate that, for the binary task of audio-based sentiment analysis, the unconventional DEEP SPECTRUM features perform slightly better than the standard and robust MFCCs features [222, 223]. The best overall classification result is obtained using DEEP SPECTRUM features quantised with $cs = 500$ and $cw = 1$. For the MFCC features, the best results are observed for a codebook with $cw = 10$, i. e. when

Table 9.5: Comparison of the classification results using BoDF and Bag-of-MFCCs for the task of audio-based sentiment analysis. Different combinations of codebook size (*cs*) and multi-assignment degree, or codebook words (*cw*) are evaluated. The SVM's cost parameter *C* is optimised on the development (Devel.) partition. The chance level is 50.0 % UAR. The best result for each feature set is highlighted with a light grey shading.

UAR [%]		$cw = 1$			$cw = 10$			$cw = 20$			$cw = 50$		
Feature Space	*cs*	*C*	Devel.	Test	*C*	Devel.	Test	*C*	Devel.	Test	*C*	Devel.	Test
Bag-of-MFCCs	2 000	10^{0}	71.7	68.1	10^{-1}	68.6	70.3	10^{-1}	67.7	70.3	10^{-4}	69.3	62.9
Bag-of-MFCCs	4 000	10^{-2}	71.9	72.2	10^{-2}	68.9	72.2	10^{-2}	68.9	69.8	10^{-2}	67.2	69.8
Bag-of-MFCCs	8 000	10^{-2}	71.2	68.9	10^{-2}	71.7	**73.1**	10^{0}	69.3	66.5	10^{-5}	66.2	67.7
BoDF	500	10^{-4}	69.5	**74.5**	10^{-6}	74.3	71.0	10^{-5}	74.3	70.5	10^{-5}	69.5	71.0
BoDF	1 000	10^{-2}	74.5	55.8	10^{-4}	74.8	68.6	10^{-5}	74.3	70.5	10^{-6}	71.9	69.8
BoDF	2 000	10^{-4}	75.3	64.1	10^{-4}	70.7	68.6	10^{-5}	73.1	70.5	10^{-5}	71.9	71.0
BoDF	4 000	10^{-3}	69.5	64.5	10^{-4}	70.7	72.2	10^{-5}	71.9	70.5	10^{-5}	71.9	71.0

creating histogram representations from a codebook with the *cs* between 2 000 and 8 000, ten suitable vectors are found for each feature vector from the quantised space. However, no assumption can be made about the codebook size *cs* within the mentioned range.

9.2.3 Conclusions

Using an existing corpus of movie reviews collected from YouTube, the suitability of the BoDF proposed by the author and his colleagues [88] and state-of-the art Bag-of-MFCC features for audio-based sentiment analysis were analysed. Presented results indicate that the DEEP SPECTRUM based systems [16, 88] consistently outperform the equivalent MFCC system. Given the strong performance of DEEP SPECTRUM features in the related task of emotion classification [4], this finding is not unexpected. This result adds to the growing evidence in the literature that feeding spectrogram representations through pre-trained image CNNs produces salient features suitable for audio classification tasks.

Future work could include fusing information from the different modalities present in the Movie Review Dataset to further improve on the DEEP SPECTRUM results, as well as performing cross-corpus multimodal sentiment analysis using reviews of different genres and cultures. It may also be beneficial to fuse the deep representations with expert-designed features developed for the INTERSPEECH's emotion challenge, as this feature set has been shown to be highly effective in capturing emotion related features [198, 199].

10

Health Monitoring

The human body produces a wide range of acoustic sounds that directly and indirectly reflect developments and changes in our physical and mental states. Such acoustic data is complex and rare in nature, exhibiting strong interconnections. In this chapter, the suitability of the deep learning methodologies proposed by the author and his colleagues, for learning meaningful representations from scarce medical data will be analysed. In particular, in Section 10.1, the proposed *Sequence to Sequence Autoencoder* (S2SAE) (cf. Chapter 6) will be applied for classification of abnormal heart sounds [163], and in Section 10.2, the DEEP SPECTRUM system (cf. Chapter 4) will be utilised to classify various snore sounds [18].

10.1 Abnormal Heart Sound Classification

Computer audition techniques have the potential to produce supporting technologies for cardiologists and general practitioners to help increase the clinical efficacy of auscultation, thus helping to reduce the high societal burden associated with heart diseases [224]. Moreover, advances in mobile and wearable recording and sensing devices are increasing the reliability and feasibility of remote diagnostic and monitoring solutions [225].

In this section, it will be investigated whether state-of-the-art computer audition paradigms can be applied to classify heart sounds. In particular, the suitability of the introduced S2SAE (cf. Chapter 6) [17, 18] for the mentioned task will be explored [163]. The performance of this RNN-based system will be compared with two non-deep approaches, a conventional acoustic feature set [226] and a *Bag-of-Audio-Words* (BOAW) approach [227].

All three approaches are verified on the *Heart Sounds Shenzhen* (HSS) corpus, a novel database of 422.82 minutes of heart sound recordings collected from 170 participants (cf. Section 10.1.1.1).

Table 10.1: Class distribution per partition.

Partition	normal	mild	moderate/severe	$\sum$
Train	84	276	142	502
Devel.	32	98	50	180
Test	28	91	44	163

10.1.1 Data and Procedure

The experiments are performed in three steps on the HSS corpus which has recently been made available through the 2018 edition of the INTERSPEECH COMPARE [164]. First, power spectrograms are generated from heartbeat recordings. Subsequently, recurrent S2SAEs are used for unsupervised representation learning from the extracted mel-spectrograms. Finally, a classifier is trained on the learnt representations to predict the labels of the heartbeat recordings. The obtained results are also compared with two other feature sets.

10.1.1.1 Heart Sound Dataset

The HSS corpus contains 845 recordings (with 30 seconds on average) representing 422.82 minutes. The heart recordings were collected using the electronic stethoscope from one of four locations: (i) the auscultatory mitral area, (ii) the aortic valve auscultation area, (iii) the pulmonary valve auscultation area, and (iv) the auscultatory area of the tricuspid valve. The recordings were collected from 170 independent subjects (55 female and 115 male), from mostly older individuals (ages range from 21 to 88 with the mean age being 65.4 years, and standard deviation of 13.2 years) with varying health conditions, including coronary heart disease, heart failure, arrhythmia, hypertension, hyperthyroid, and valvular heart disease [163].

The corpus has been categorised into three classes: (i) normal, (ii) mild, and (iii) moderate/severe, as diagnosed by specialists in heart diseases. These classes are divided into participant-independent training, development, and test sets with 502, 180, and 163 audio instances, respectively (cf. Table 10.1). The gender and age classes are evenly distributed.

10.1.1.2 Spectrogram Creation

First, the power spectra of heartbeat audio samples are generated using periodic Hanning windows of width w ms and overlap $0.5\,w$ ms. Subsequently, N_{mel} log-scaled mel-frequency bands are computed from the spectra. These representations have previously been shown to be effective for heart sound classification [228]. Finally, the mel-spectrograms are normalised in $[-1,1]$, as the outputs of the autoencoder are constrained to this interval. Furthermore, important acoustic cues related to the class label may be

obscured by background noise during the recording of the heartbeats. Hence, it is investigated whether removing some background noise from the spectrograms improves system performance. This is achieved by clipping amplitudes below four certain thresholds, $-30\,\mathrm{dB}$, $-45\,\mathrm{dB}$, $-60\,\mathrm{dB}$, and $-75\,\mathrm{dB}$ [163].

10.1.1.3 Recurrent Autoencoders

The detailed description of the applied S2SAE is given in Chapter 6. The Adam optimiser is used to train the autoencoders with an initial learning rate of 0.001 [182] for 64 epochs in batches of 256 samples. A dropout of 20 % has been applied to the outputs of each recurrent layer [38]. Moreover, gradients with absolute value above 2 are clipped [157]. As described in Chapter 6, suitable values for the hyperparameters are selected in three stages.

First, the number of recurrent layers N_{layer}^{S2SAE} and *Gated Recurrent Units* (GRUs) N_{unit}^{S2SAE} in each RNN layer are optimised with $N_{layer}^{S2SAE} \in \{2,3,4\}$, $N_{unit}^{S2SAE} \in \{64,128,256,512\}$. All combinations of bidirectional or unidirectional encoder and decoder RNNs are evaluated. The highest *Unweighted Average Recall* (UAR) was achieved when using $N_{layer}^{S2SAE} = 2$ and $N_{layer}^{S2SAE} = 256$ GRUs with a unidirectional encoder RNN and a bidirectional decoder RNN.

Second, using the autoencoder configuration specified in the first stage, the window width w for spectrogram creation is evaluated between 80 and 360 ms with a step size of 40 ms. It was observed that the windows size $w = 320\,\mathrm{ms}$ provided the strongest UAR. It is speculated that using window sizes shorter than $w < 320\,\mathrm{ms}$ result in weaker representation due to the lack of discriminating information in the shorter audio segments. For larger values of $w > 320\,\mathrm{ms}$, it was observed that the classification accuracy dropped again. This could have been caused by the larger window width blurring the short-term dynamics of the heartbeat sounds.

In the final optimisation stage, various numbers of mel-frequency bands $N_{mel} \in \{16,32,64,128,256\}$ were tested. With larger values of N_{mel}, the UAR rises until it stops increasing for $N_{mel} > 128$. For this reason, $N_{mel} = 128$ is chosen to reduce the amount of data which the system has to process.

10.1.1.4 ComParE Acoustic Feature Set

Further results presented are based on the INTERSPEECH 2016 ComParE feature set [229]. This feature set includes a range of prosodic, spectral, cepstral, and voice quality LLDs contours, to which statistical functionals such as the mean, standard deviation, percentiles and quartiles, linear regression descriptors, and local minima/maxima related descriptors are applied to produce a 6 373 dimensional static feature vector. For the full description of this feature set, the interested reader is referred to [230].

10.1.1.5 Bag-of-Audio-Words

Bag-of-Audio-Words (BoAW), which have been computed using the toolkit openXBOW [227], are also tested. BoAW involve the quantisation of acoustic LLDs to form a sparse fixed length histogram (bag) representation of an audio clip.

All BoAW representations were generated from the 65 LLDs and corresponding deltas in the ComParE feature set (cf. Section 10.1.1.4) [230]. Prior to quantisation, the LLDs were normalised to zero mean and unit variance. All codebooks were learnt using openXBOW random sampling setting with codebook size (*cs*) 250, 500, and 1 000.

10.1.2 Results

In order to predict the class labels for the audio instances in the HSS corpus, a linear SVM classifier is trained using the *Sequential Minimal Optimization* (SMO) algorithm [231] implemented in Weka 3.8.2 [221]. Features were scaled to zero mean and unit variance, using the parameters from the training set. The complexity hyperparameter of the SVM (C) is optimised on a logarithmic scale between 10^{-6} and 10^{-1} with a factor of 10 for the deep learning, ComParE, and BoAW approaches. The SVM complexity that performs the strongest on the development set is applied to train the final classifier with the fusion of the training and development sets. For the BoAW results on the test set, the codebook is learnt again from the fused data. Due to small imbalances in the class distribution of the data (cf. Table 10.1), all classification systems are evaluated using the UAR metric.

The strongest development set UAR, 50.3 % (cf. Table 10.2), was achieved using a system based on the ComParE feature set and a SVM complexity of $C = 10^{-4}$. However, this system had a noticeable drop in performance on the HSS test partition indicating possible overfitting. For the conventional (non-deep) approaches, the strongest test set partition UAR, 47.2 % (cf. Table 10.2) was achieved using a BoAW approach with $cs = 500$, and $C = 10^{-3}$.

For the deep recurrent approach, the learnt representations achieved a weaker performance than the conventional feature sets on the development partition. This could be due to the small amount of data for training the autoencoders. When comparing with the conventional approaches on the HSS test set, the learnt representations achieve equivalent performance. Moreover, an early fusion of the four learnt deep feature vectors obtain the highest UAR, 47.9 % (cf. Table 10.2) on the test set. This result indicates the promise of deep representation learning for abnormal heart sound classification.

10.1.3 Conclusions

Technologies based on state-of-the-art computer audition systems have the potential to aid the diagnosis of cardiovascular disorders. In this regard, the presented results indicated the potential of deep learning to learn meaningful representations from *Phonocardiogram* (PCG) recordings. It was shown that fusing all deep representations after amplitude

Table 10.2: A comparison of the UARs of the S2SAE system with a ComParE feature set and a BoAW approach. The chance level is 33.3 % UAR. The best result is highlighted with a light grey shading.

System	Dimensionality	UAR [%] C	Devel.	Test
ComParE	6 373	10^{-6}	41.1	44.8
		10^{-5}	44.5	45.6
		10^{-4}	50.3	46.4
		10^{-3}	44.5	40.4
		10^{-2}	43.2	41.7
BoAW	250	10^{-3}	43.1	43.4
	500	10^{-3}	42.3	47.2
	1 000	10^{-2}	43.7	41.0
S2SAE: Individual Feature Sets				
-30 dB	1 024	$2 \cdot 10^{-2}$	32.8	40.0
-45 dB	1 024	$5 \cdot 10^{-4}$	38.4	40.6
-60 dB	1 024	$6 \cdot 10^{-2}$	39.6	45.2
-75 dB	1 024	$8 \cdot 10^{-3}$	36.9	41.7
Fused	4 096	$4 \cdot 10^{-3}$	35.2	**47.9**

clipping, it is possible to outperform the conventional acoustic features for the task of abnormal heart sound classification. In future work, PCG databases assembled for the *Computing in Cardiology* (CinC) Challenge 2016 [232] should be considered to provide further training material for the S2SAE.

10.2 Snore Sound Recognition

Snoring can be a marker of *Obstructive Sleep Apnea* (OSA) [233] which, after insomnia, has the highest prevalence of all sleep disorders, affecting approximately 3–7 % of the middle-aged men and 2–5 % of the middle-aged women [234–236] in the general population. OSA is characterised by repetitive episodes of partial, or complete collapses of the upper airway during sleep, causing impaired gaseous exchanges and sleep disturbance [237]. OSA can lead to an increased risk of cardiovascular and cerebrovascular diseases [238,239]. An integral part of successful treatment is locating the site of obstruction and vibration [240], which was the subject of the INTERSPEECH 2017 COMPARE Snoring sub-challenge [5]. The challenge requires participants to identify four different sources of vibration from audio snore samples: epiglottis (E), oropharyngeal lateral walls (O), tongue (T), and velum (V).

An audio perspective on the analysis of snoring has made use of, among others, amplitude [241], frequency [242], and wavelet features [243]. In this section, the applicability of DEEP SPECTRUM (cf. Chapter 4) to extract robust representations for the task of snore sound recognition will be analysed [16].

10.2.1 Data and Procedure

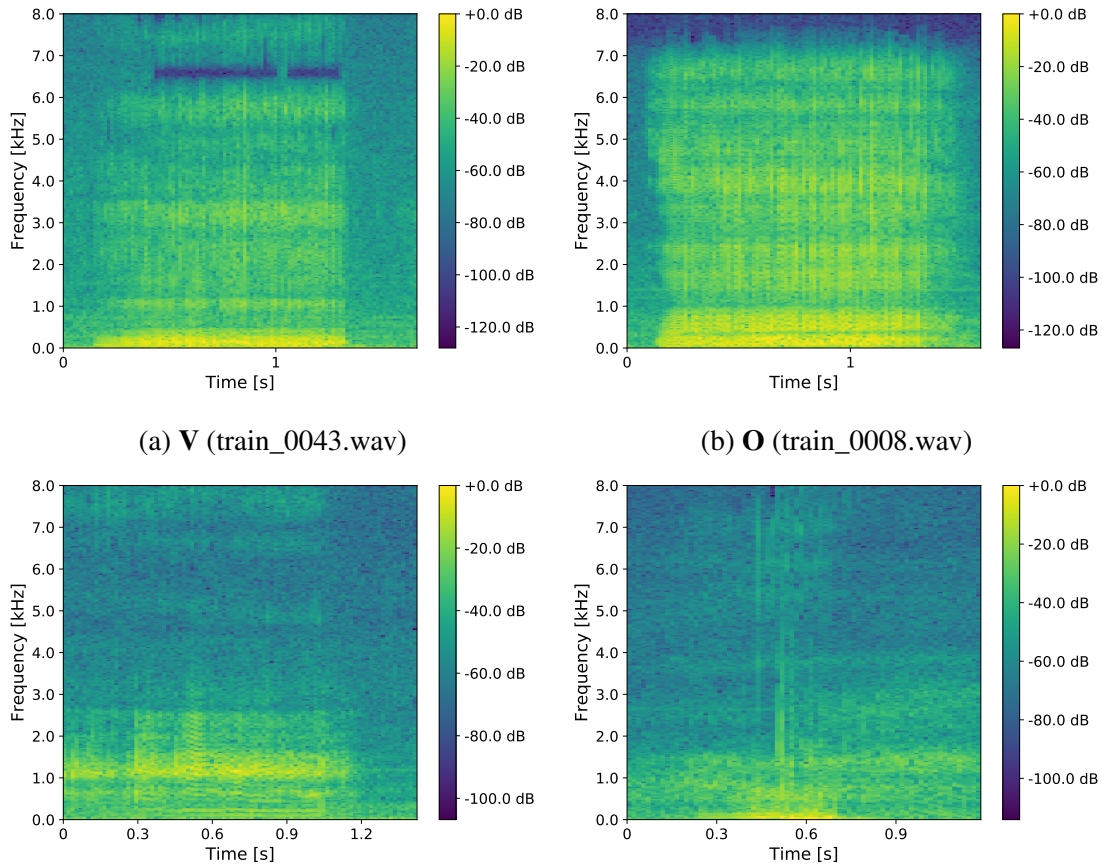
For the deep learning experiments the *Munich-Passau Snore Sound Corpus* (MPSSC) has been used. In the first experimental stage, spectrograms are extracted from each snore sound (cf. Figure 10.1). Subsequently, AlexNet and VGG19 are used to extract the DEEP SPECTRUM features (cf. Section 10.2.1.3). After obtaining the deep representations, early and late fusion experiments have been performed (cf. Section 10.2.1.4). Finally, the obtained results are analysed and compared with the provided baselines in Section 10.2.2.

10.2.1.1 Munich-Passau Snore Sound Corpus

The INTERSPEECH 2017 COMPARE Snoring sub-challenge is based on the MPSSC, which contains 828 snore samples from four classes. Each one of these classes relates to one source of vibration. For the challenge, the corpus has been split equally into training, development, and test partitions [5]. The classes have uneven distribution, with substantially more *V* samples (cf. Table 10.3). Therefore, upsampling of the data is performed, by replicating samples from the *O*, *T*, and *E* classes proportional to their relative frequency. The same upsampling factors as those used in the challenge baseline system are applied. This results in all classes having approximately the same number of audio recordings. For a detailed description of the corpus and the class distributions the reader is referred to [5].

Table 10.3: Class distribution per partition. V: velum, O: oropharyngeal lateral walls, T: tongue base, E: epiglottis.

Partition	V	O	T	E	∑
Train	168	76	8	30	282
Devel.	161	75	15	32	283
Test	155	65	16	27	263



(a) **V** (train_0043.wav)

(b) **O** (train_0008.wav)

(c) **T** (train_0087.wav)

(d) **E** (train_0042.wav)

Figure 10.1: Representative spectrograms for the different types of snore sounds using the best performing colour map *viridis*. Each one of the four classes relating to the point of vibration (V: velum, O: oropharyngeal lateral walls, T: tongue base, E: epiglottis) produces a unique spectral image. The samples from which these spectrograms have been extracted are given in parentheses.

10.2.1.2 Spectrograms Creation

Hanning windows of width $w = 16$ ms and overlap 8 ms are used, and the power spectral density on the dB power scale is computed. The impact of three 'standard' spectrogram colour mappings is analysed to find a good representation of the snore samples: *jet* which is the default colour map of matplotlib [220] and varies from blue (low range) to green (mid range) to red (upper range); *gray* which is a sequential grey-scale mapping which varies from black (low range) to grey (mid range) to white (upper range); and finally, *viridis* which is a perceptually uniform sequential colour map, varying from blue (low range) to green (mid range) to yellow (upper range). Example plots as used by the final system for each class of the Snoring sub-challenge are shown in Figure 10.1. It is worth noting that, even with the human eye, some clear distinctions between the spectrograms of different classes can be made.

10.2.1.3 DEEP SPECTRUM Features

At this stage, the DEEP SPECTRUM features are evaluated for all combinations of CNN-descriptors and spectrogram colour maps, resulting in 12 different configurations (cf. Table 10.4). Features are extracted from spectrograms of three different colour maps by using *fc6*/*fc7* activations of both AlexNet and VGG19 (cf. Section 4.2.2). The LibLINEAR library with the L2-regularised L2-loss dual solver [212] is used via the Weka machine learning toolkit [221], and the SVM complexity parameter $C \in [10^{-6}, 10^{-1}]$ is optimised on the development partition. For each configuration, two best results of adjusting C are presented. The performance of the configurations is scored using UAR, as specified in the challenge baseline [5].

10.2.1.4 Fusion

Finally, the effects of three different fusion scenarios on the proposed system are evaluated: First, the DEEP SPECTRUM features extracted from the spectrograms of the different colour maps are fused to investigate whether a specific mapping contains important information that cannot be found in the others. Second, fusion of the different CNN layers used for feature extraction is performed. Third, descriptors of different CNN architectures are fused to analyse whether they complement each other. Therefore, there are three models left which have to be evaluated: (i) fusing features extracted from AlexNet's *fc7* layer for all three colour maps (*Colour-Map Fusion*), (ii) combining the features extracted from both of AlexNet's fully connected layers *fc6* and *fc7* (*Layer Fusion*), and (iii) fusing features extracted from *fc7* of AlexNet and VGG19 (*CNN Fusion*). The best performing colour map *viridis* is used for layer and CNN fusion [16].

In all of these scenarios, both feature and decision-level fusions are evaluated. Feature level fusions are performed by concatenation and classification via linear SVM, and decision-level fusions by majority voting of the best non-fused linear SVM configurations, i. e. the optimal value for C determined during development is used.

Table 10.4: Results for the snore sub-challenge using linear SVM on four different CNN-descriptors (AlexNet *fc6*, AlexNet *fc7*, VGG19 *fc6*, and VGG19 *fc7*) extracted from spectrograms with three different colour maps (*gray*, *jet*, and *viridis*). UAR % is used as measure and C is optimised on the development partition. The chance level is 25.0 % UAR. On the test set, only five trials were allowed to be uploaded. The best result on the test set is highlighted with a light grey shading.

		UAR [%]				UAR [%]	
CNN-descriptor	C	Devel.	Test	CNN-descriptor	C	Devel.	Test
AlexNet *fc6*				AlexNet *fc7*			
gray	10^{-1}	39.7	–	gray	10^{-1}	38.2	–
	10^{-3}	42.0	–		10^{-2}	38.2	–
jet	10^{-4}	36.2	–	jet	10^{-2}	37.4	–
	10^{-6}	36.8	–		10^{-4}	38.8	–
viridis	10^{-4}	43.5	–	viridis	10^{-3}	47.4	63.3
	10^{-5}	41.6	–		10^{-4}	44.8	**67.0**
VGG19 *fc6*				VGG19 *fc7*			
gray	10^{-3}	28.4	–	gray	10^{-2}	29.9	–
	10^{-4}	30.7	–		10^{-3}	31.5	–
jet	10^{-2}	31.2	–	jet	10^{-1}	31.7	–
	10^{-3}	31.4	–		10^{-2}	31.3	–
viridis	10^{-4}	38.5	–	viridis	10^{-2}	39.5	–
	10^{-5}	37.4	–		10^{-3}	39.0	–

Table 10.5: Comparison of fusion strategies for the DEEP SPECTRUM system. Different colour mappings, layers, and CNN architectures all on both feature (feat.) and decision (dec.) level are fused. Linear SVM is used for feature-level fusion (optimising C on the development partition) and the best SVM models obtained during development of the non-fusion configurations are used. For a detailed description of which colour maps, layers, and CNN architectures are fused the reader is referred to Section 10.2.1.4.

Fusion Model	UAR [%]			
	devel		test	
	feat.	dec.	feat.	dec.
Colour-Map Fusion	38.1	42.2	–	–
Layer Fusion	43.8	46.1	–	63.8
CNN Fusion	44.7	46.4	57.4	62.0

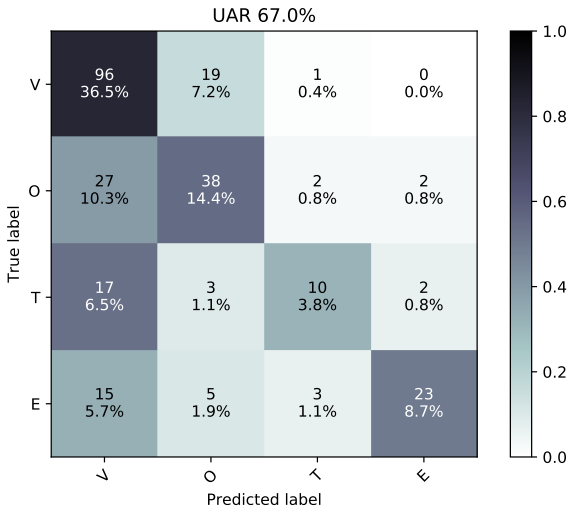


Figure 10.2: Confusion Matrix of the best classification on the test set instances using AlexNet's *fc7* and *viridis* as colour map. *V*: velum, *O*: oropharyngeal lateral walls, *T*: tongue base, *E*: epiglottis.

10.2.2 Results

The results of the experiments, across all 12 combinations of spectrogram colour maps, pre-trained CNNs, and extraction layer used to form the different deep spectrum representations, are shown in Table 10.4. Best results are achieved with features extracted from AlexNet's *fc7* layer and the spectrogram colour map *viridis*. With $C = 10^{-4}$, a UAR of 44.8 % is achieved on the development, and a UAR of 67.0 % on the test partition, outperforming the challenge's baseline system (cf. Table 10.6). The confusion matrix of classification labels on the test set for this best performing system is displayed in Figure 10.2.

Analysing the results produced by the different fusion configurations (cf. Table 10.5), it can be seen that fusing features extracted from spectrograms of different colour maps decreases performance for both feature and decision-level fusion compared to only using the best colour map. While fusing features extracted from different layers reduces performance for early fusion, decision-level fusion produces results similar to the respective single layer configuration. The performance decrease might be caused by the increased feature size. Lastly, fusing the best performing features from AlexNet and VGG19 produces similar results: decreased performance for feature-level fusion and performance similar to the non-fusion model for decision-level fusion.

The evaluation also shows two characteristics of interest from the extracted DEEP SPECTRUM features. First, the features extracted from AlexNet perform better than those of VGG19. This is the opposite of the results presented for the ImageNet task, in which VGG19 drastically outperforms AlexNet [30]. Second, the choice of colour map in the spectrogram creation step has an observable impact on the performance of the entire

Table 10.6: Comparison of the DEEP SPECTRUM based approach with the challenge baseline (functionals), and the end-to-end approach (CNN & LSTM) used in the baseline paper. The best result on the test set is highlighted with a light grey shading.

Model	Ref.	UAR [%] Devel.	Test
Baseline CNN & LSTM	[5]	40.3	40.3
Baseline functionals	[5]	40.6	58.5
Baseline COMPARE BOAW + SVM	[5]	44.2	51.2
Baseline (Late fusion)	[5]	43.4	55.6
Deep Spectrum	Table 10.4	**44.8**	**67.0**

system: in all but one configuration (AlexNet *fc6*) *viridis* increases UAR, while a simple grey-scale mapping leads to improvements over the standard *jet* map only for AlexNet *fc6*. Since both nets are predominantly trained on a large corpus of natural images, it seems to be intuitive that the choice of colour map for an artificial image like a spectrogram plot would impact the models' ability to extract useful features.

10.2.3 Conclusions

This work proposed a method for classifying snore sounds that relies on the ability of large, deep pre-trained CNNs to extract useful information from spectrograms. Using the DEEP SPECTRUM system for feature extraction, and linear SVM as a classifier, it was possible to substantially outperform the baseline for the snoring sub-challenge which utilises classic knowledge-based audio features. In comparison to the baseline features, the DEEP SPECTRUM system relies solely on spectral information and large, deep CNNs' ability to infer a higher level representation of arbitrary input images. In the experiments, it was found that both the choice of colour map for the spectrogram plots, and the pre-trained CNN used for feature extraction have a substantial impact on performance. Further research should investigate segmenting the audio files into chunks of equal length prior to generating the spectrograms, in this way providing input, for example for recurrent networks with LSTM cells. It might also be beneficial to fuse the DEEP SPECTRUM representations with conventional acoustic feature sets, as such features have had success in the past [164, 198, 199, 226, 244, 245].

Part V

Concluding Remarks

11

Concluding Remarks

11.1 Summary

The research presented in this thesis was oriented towards addressing four scientific objectives, which were formulated based on the existing challenges in the field of audio processing (cf. Section 1.3): I) whether pre-trained image classification CNNs can extract meaningful representations from audio spectrograms, II) whether quantising deep features can efficiently reduce noisy representations in the feature space, III) whether DCGANs and autoencoders are suitable for unsupervised representation learning from audio data, and IV) whether both shift-invariant, high-level features, and the long-term temporal context of audio signals can be captured with CRNNs.

The first objective was addressed by introducing the DEEP SPECTRUM system, with which it was possible to learn highly robust and meaningful representations from audio data and achieve state-of-the-art results for challenging audio tasks, including game genre classification (cf. Section 8.3), in-the-wild speech and vocalisations classification (cf. Section 9.1), and snore sound recognition (cf. Section 10.2). The results also indicated that the CNNs utilised in DEEP SPECTRUM, despite being exclusively pre-trained on the ImageNet corpus [73], are highly effective for learning representations from audio.

To validate the second objective, the author and his colleagues introduced the BODF, which are fixed length histogram (quantised) representations of the time-continuous DEEP SPECTRUM features. The feature quantisation process can be considered as a low-pass filtering operation, which effectively compresses the feature space and eliminates noisy representations. In Sections 8.3 and 9.1, it was found that using BODF, it is possible to get stronger representations than the DEEP SPECTRUM and other competitive feature sets for a variety of audio recognition tasks. In Section 9.2, the BODF showed higher performance in comparison with the state-of-the-art Bag-of-MFCC features for audio-based sentiment analysis.

For the third objective, a DCGAN structure and a S2SAE were introduced. Both of these approaches are completely unsupervised, hence eliminating the need for expert-designed features. For the DCGAN, the activations of the discriminator, and for the S2SAE, the activations of the fully connected layer were extracted as representations of the input audio signals. It was then shown that using these representations, it is possible to achieve state-of-the-art results for the task of acoustic scene classification (cf. Section 8.1). It was also found that the fusion of the prediction probabilities obtained from the DCGAN and S2SAE can further improve the results in a classification paradigm. Furthermore, the results in Section 10.1 indicate the suitability of the S2SAE for medical data.

The final objective was addressed by showing that the introduced CRNN architecture is able to achieve state-of-the-art results for the task of rare acoustic event detection (cf. Section 8.2). It could also be demonstrated that the inclusion of LSTM cells leads to more accurate onset predictions and the results indicated that long term temporal properties are efficiently modelled, compared to the individual architectures using MLPs or a CNN with a FFNN (cf. Section 8.2).

11.2 Limitations

Although the deep learning approaches introduced throughout this thesis are highly effective as a means of learning valuable representation from a wide variety of audio data, there are three major limitations that should be highlighted for these methods.

Lack of transparency in deep architectures. In recent years, the relative opacity of DNNs has been a major discussion topic [246, 247]. To what extent this issue matters in the long term remains an open question [248], which can be addressed depending on the application area and the robustness of the utilised DNNs [249]. This question may not be of high relevance, as long as a deep computer audition system, for example for the task of game genre classification (cf. Section 8.3), is self-contained and performs robustly. However, the opacity of DNNs becomes problematic in various decision-making processes related to ethical, legal, and quality control issues, for example, in the event of misclassification of an abnormal heart sound (cf. Section 10.1) which may have serious health implications for the affected patient. Currently, there is no specific way to fully understand or trace back the reasoning behind the decision made by DNNs. Furthermore, to solve more complex tasks, deeper networks (e. g. by adding more hidden layers and neurons to the network) have been developed [144, 214, 250]. With huge degrees of freedom and high complexity of these networks, the quality of being debuggable deteriorates, which makes it almost impossible for non-experts to analyse the networks structure, gradients, weights, and interconnections between neurons.

The need for more data. Unlike human beings, who are able to learn abstract relation-

ships from only a few trials, DNNs lack the mechanism to learn abstractions through explicit limited definitions [249]. Instead, these networks work best when trained with thousands or even millions of data samples. Lake et al. emphasised in various works, that humans are much more efficient at learning complex rules as compared to DNNs [251, 252]. This statement was substantiated in Section 8.3, as a human annotator – without having access to the training data, and in much less time – has achieved a classification UAR almost as good as the best performing machine learning system, which was a DNN. To address problems where the training data is limited, e. g. for the abnormal heart sound classification task described in Section 10.1, the author has demonstrated that unsupervised representation learning with S2SAEs can help to prevent overfitting [163]. However, it should be indicated that the contemporary non-deep approaches showed almost equivalent performance as the applied S2SAE, and consume less computational time and resources. In various experiments throughout this thesis, a common pattern can be seen: where the training data is scarce, the performance of DNNs is almost similar to (or just slightly better than) non-deep approaches (e. g. see results in Sections 8.3, 9.1, 9.2 and 10.1).

Hyperparameter search. Representation learning systems, e. g. the introduced DCGAN (cf. Chapter 5) and S2SAE (cf. Chapter 6) contain a wide range of adjustable hyper-parameters, which prohibits an exhaustive analysis of the parameter space. Exploring the architectures of such deep systems and finding good performing configurations is an art. In case of S2SAE, practical experiences with autoencoders and a certain amount of theoretical knowledge in the field of audio signal processing is required to set parameters for: i) spectrogram creation: window width and number of mel-filterbanks, and ii) autoencoder training: number of recurrent layers, number of recurrent units, type of recurrent cells, direction of the decoder and encoder RNNs, learning rate, and dropout.

The above mentioned limitations were the most related ones to the DNN architectures introduced throughout this thesis. A broader and a more general discussion related to this topic is provided in [249].

11.3 Outlook

With respect to the limitations mentioned above, the DNNs should be reconceptualised, i. e. not be considered as the only technique to solve machine learning problems, but more as an effective tool amongst other possible techniques. Having said this, however, conventional machine learning systems often reach their limit for solving more complex computer audition problems, such as speech processing or rare event detection (cf. Section 8.2). Hence, as indicated, deep structures are crucial, but with the premise that there is enough data available to train such system (cf. Section 11.2). For the future work, following possibilities should be considered more intensively.

Attention-based learning mechanism. An attention paradigm, which is originally inspired by the visual attention mechanism found in humans, decides which part of the input signals should be used to generate output, instead of processing the entire sequences of input. Following Graves [253] who proposed the attention mechanism to build a neural network that creates convincing handwriting from a given text, and with respect to the attention-based neural machine translation model proposed by Bahdanau et al. [254], future works should consider building an attention mechanism in the introduced S2SAE (cf. Chapter 6). This means, that the decoder RNN decides on which part of the input sequence it should focus. With this attention mechanism, the encoder RNN is relieved from encoding all information in the whole input sequence of spectrograms (cf. Section 6.2) into a fixed-length vector. Having this mechanism integrated within the introduced S2SAE, the network might then be able to process longer and noisier inputs, whilst extracting more robust representations.

Holistic audio understanding. Motivated by the humans auditory sensors, future computer audition systems should be able to obtain a higher level semantic understanding across multiple audio domains, including acoustic environment, speech, music, and other sound events. This degree of understanding goes beyond the detection of an audio event (cf. Section 8.2) or recognition of various audio classes as well as their respective attributes that appear in an audio segment (e. g. classification paradigms discussed in Sections 8.1 and 9.1). The ideal goal should be to understand the inter-relations between each event (e. g. by creating an ontology tree), and evaluate their relative importance (e. g. by utilising an attention mechanism). Schuller [255] drew the vision of holistic evolutionary audio understanding to simultaneously analyse a real-life audio stream in terms of recognising various speakers and their states and traits, acoustic environments, as well as other sound sources. For realisation of this vision, a combination of an audio decomposition component and unsupervised representation learning techniques could be highly beneficial. The audio decomposition component would then have three main tasks: i) source separation, i. e. decomposing input audio stream into separate different audio sources, e. g. by using NMF or *Non-negative Tensor Factorisation* (NTF) [256–259], ii) speaker diarisation, i. e. the process of tagging the input audio of various speakers with each speaker's turn information to determine 'who is speaking when', e. g. by utilising RNNs [260, 261], and iii) audio diarisation, which is a higher level abstraction of speaker diarisation to other sound sources, e. g. background noise types or musical instruments [262]. Afterwards, unsupervised representation learning methods, such as S2SAEs can be applied to extract valuable features from decomposed audio segments [18]. These representations can then be sent to a classifier to predict the labels. Finally, an interpreter component can put the label information together and prepare the system output.

Big data and model robustness. In order to increase the robustness of machine learning

models, future work should also aim to harness the so called five Vs (*value, variety, velocity, veracity,* and *volume*) of the big data era [263–265]. Real-world data sourced from social multimedia (e. g. *AudioSet* [93] for audio and *YouTube-8M* [215] for video) should be used to provide *volume* and *variety*. To check the *veracity* of the data, cooperative and semi-supervised active learning algorithms can be applied [266, 267]. Such large quantities of data can advance research in (unsupervised) representation learning, domain adaptation, or training noise robust models. In recent years, deep learning has shown to be highly effective for such big data dimensions. In particular, to learn representations from data of various modality, Ngiam et al. [268] and Srivastava et al. [269] introduced novel multimodal representation learning approaches, which should be further considered for future work. To address the *Velocity* factor, there is a need for machine learning systems for large-scale processing of the mass of produced data. Such a system should be able to extract valuable features in a short time period. This thesis contributes to a realisation of this factor by introducing DEEP SPECTRUM, which facilitates process parallelisation for rapid GPU-based deep feature extraction (cf. Chapter 4). Finally, the last and probably the most important V that should be taken into account when working with big data is its *Value*, i. e. using the collected data to solve real-world problems and help towards building robust computer audition models for holistic audio understanding. In this way, a good starting point would be to introduce more challenges similar to the ImageNet Large Scale Visual Recognition Challenge [270], but instead aimed at solving audio classification problems.

Acronyms

ACS Action or Shooter

AI Artificial Intelligence

ANN Artificial Neural Network

ARP Arcade or Platform

ASC Autistic Spectrum Condition

ASR Automatic Speech Recognition

AVEC Audio/Visual Emotion Challenge and Workshop

BLSTM Bidirectional Long Short-Term Memory

BN Batch Normalisation

BoAW Bag-of-Audio-Words

BoDF Bag-of-Deep-Features

BPTT Backpropagation Through Time

CAS^{2}T Cost-efficient Audio-visual Acquisition via Social-media Small-world Targeting

CCA Canonical Correlation Analysis

CFS Correlation-based Feature Selection

CinC Computing in Cardiology

CNN Convolutional Neural Network

ComParE Computational Paralinguistics ChallengE

CRNN Convolutional Recurrent Neural Network

CSO Competitive Swarm Optimisation

CV Cross Validation

DCASE Detection and Classification of Acoustic Scenes and Events

DCGAN Deep Convolutional Generative Adversarial Network

DNN Deep Neural Network

DSD Deceptive Speech Database

EBR Event to Background Ratio

EPP Event Presence Probability

ER Error Rate

FFT Fast Fourier Transform

FFNN Feedforward Neural Network

FHT Fighting

GAN Generative Adversarial Network

GEMEP Geneva Multimodal Emotion Portrayals

GMM Gaussian Mixture Model

GPU Graphics Processing Unit

GRU Gated Recurrent Unit

HMM Hidden Markov Model

HNR Harmonics-to-Noise Ratio

HSS Heart Sounds Shenzhen

ICA Independent Component Analysis

ILSVRC ImageNet Large Scale Visual Recognition Challenge

InfoGANs Information Maximising Generative Adversarial Networks

*k***NN** *k*-Nearest-Neighbour

LLD Low-Level Descriptor

LSTM Long Short-Term Memory

MFCC Mel-Frequency Cepstral Coefficient

MI Mutual Information

MLP Multilayer Perceptron

MPSSC Munich-Passau Snore Sound Corpus

MPC Mixtures Per Class

MRMR Minimum-Redundancy-Maximal-Relevance

MSE Mean Squared Error

NLP Natural Language Processing

NMF Non-negative Matrix Factorisation

NTF Non-negative Tensor Factorisation

OSA Obstructive Sleep Apnea

PCA Principal Component Analysis

PCG Phonocardiogram

PCM Pulse Code Modulation

RBM Restricted Boltzmann Machine

RCG Racing

ReLU Rectified Linear Unit

RMS Root Mean Square

RMSE Root Mean Squared Error

RNN Recurrent Neural Network

RSED Rare Sound Event Detection

S2SAE Sequence to Sequence Autoencoder

SD Standard Deviation

SED Sound Event Detection

SFM Stochastic Feature Mapping

SFS Sequential Forward Selection

SGD Stochastic Gradient Descent

SMO Sequential Minimal Optimization

SPT Sports

SVM Support Vector Machine

SWB Simulation or World Building

UAR Unweighted Average Recall

VTLP Vocal Tract Length Perturbation

ZCR Zero-Crossing-Rate

Bibliography

[1] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE 2017 challenge setup: Tasks, datasets and baseline system," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop*. Munich, Germany: IEEE, November 2017.

[2] A. Härmä, M. F. McKinney, and J. Skowronek, "Automatic surveillance of the acoustic activity in our living environment," in *Proceedings of the Int. Conf. on Multimedia and Expo*. Amsterdam, The Netherlands: IEEE, July 2005, no pagination.

[3] D. Yu and L. Deng, *Automatic speech recognition: A deep learning Approach*. London: Springer, 2016.

[4] N. Cummins, S. Amiriparian, G. Hagerer, A. Batliner, S. Steidl, and B. Schuller, "An image-based deep spectrum feature representation for the recognition of emotional speech," in *Proceedings of the 25th ACM International Conference on Multimedia, MM 2017*. Mountain View, CA: ACM, October 2017, pp. 478–484.

[5] B. Schuller, S. Steidl, A. Batliner, E. Bergelson, J. Krajewski, C. Janott, A. Amatuni, M. Casillas, A. Seidl, M. Soderstrom, 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 *Proceedings of INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association*. Stockholm, Sweden: ISCA, August 2017, pp. 3442–3446.

[6] J. Ye, T. Kobayashi, and T. Higuchi, "Audio-based indoor health monitoring system using flac features," in *2010 International Conference on Emerging Security Technologies*, September 2010, pp. 90–95.

[7] T. E. Taylor, Y. Zigel, C. De Looze, I. Sulaiman, R. W. Costello, and R. B. Reilly, "Advances in audio-based systems to monitor patient adherence and inhaler drug delivery," *Chest*, 2017.

[8] F. Eyben, F. Weninger, F. Gross, and B. Schuller, "Recent developments in opensmile, the munich open-source multimedia feature extractor," in *Proceedings of ACM Multimedia*, ACM. Barcelona, Catalunya, Spain: ACM, 2013, pp. 835–838.

[9] F. Eyben, M. Wöllmer, and B. Schuller, "openSMILE: The Munich versatile and fast open-source audio feature extractor," in *Proceedings of the 18th ACM international conference on Multimedia*. Firenze, IT: ACM, 2010, pp. 1459–1462.

[10] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[11] F. Eyben, K. R. Scherer, B. W. Schuller, J. Sundberg, E. André, C. Busso, L. Y. Devillers, J. Epps, P. Laukka, S. S. Narayanan *et al.*, "The Geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing," *IEEE Transactions on Affective Computing*, vol. 7, no. 2, pp. 190–202, 2016.

[12] F. Eyben, G. L. S. ao, J. Sundberg, K. Scherer, and B. Schuller, "Emotion in the singing voice – a deeper look at acoustic features in the light of automatic classification," *EURASIP Journal on Audio, Speech, and Music Processing, Special Issue on Scalable Audio-Content Analysis*, vol. 2015, 2015.

[13] E. Marchi, B. Schuller, S. Baron-Cohen, O. Golan, S. Bölte, P. Arora, and R. Häb-Umbach, "Typicality and emotion in the voice of children with autism spectrum condition: Evidence across three languages," in *Proceedings of INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association*. Dresden, Germany: ISCA, September 2015, pp. 115–119.

[14] L. Nanni, Y. M. Costa, A. Lumini, M. Y. Kim, and S. R. Baek, "Combining visual and acoustic features for music genre classification," *Expert Systems with Applications*, vol. 45, pp. 108–117, 2016.

[15] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[16] S. Amiriparian, M. Gerczuk, S. Ottl, N. Cummins, M. Freitag, S. Pugachevskiy, and B. Schuller, "Snore sound classification using image-based deep spectrum features," in *Proceedings of INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association*. Stockholm, Sweden: ISCA, August 2017, pp. 3512–3516.

[17] M. Freitag, S. Amiriparian, S. Pugachevskiy, N. Cummins, and B. Schuller, "audeep: Unsupervised learning of representations from audio with deep recurrent neural networks," *Journal of Machine Learning Research*, vol. 18, no. 173, pp. 1–5, 2018.

[18] S. Amiriparian, M. Freitag, N. Cummins, and B. Schuller, "Sequence to sequence autoencoders for unsupervised representation learning from audio," in *Proceedings of the 2nd Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE 2017)*. Munich, Germany: IEEE, November 2017, pp. 17–21.

[19] S. S. Haykin, S. S. Haykin, S. S. Haykin, and S. S. Haykin, *Neural networks and learning machines*. Pearson Upper Saddle River, NJ, USA:, 2009, vol. 3.

[20] 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.

[21] D. E. Rumelhart, R. Durbin, R. Golden, and Y. Chauvin, "Backpropagation: The basic theory," *Backpropagation: Theory, architectures and applications*, pp. 1–34, 1995.

[22] B. Yegnanarayana, *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.

[23] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[25] 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.

[26] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural networks*, vol. 12, no. 1, pp. 145–151, 1999.

[27] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, 2013, pp. 1139–1147.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 6, pp. 1137–1149, 2017.

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[31] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," in *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 5582–5586.

[32] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[33] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on speech and audio processing*, vol. 6, no. 2, pp. 131–142, 1998.

[34] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," *Journal of the Acoustical Society of Japan (E)*, vol. 11, no. 2, pp. 71–76, 1990.

[35] N. Jaitly and G. E. Hinton, "Vocal tract length perturbation (vtlp) improves speech recognition," in *Proceedings of ICML Workshop on Deep Learning for Audio, Speech and Language*, vol. 117, 2013.

[36] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CoRR*, vol. abs/1611.07004, 2016.

[37] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *CoRR*, vol. abs/1712.04621, 2017.

[38] N. Srivastava, G. E. 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.

[39] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012.

[40] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *International Conference on Machine Learning*, 2013, pp. 1058–1066.

[41] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[42] G.-B. Huang and H. A. Babri, "Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions," *IEEE Transactions on Neural Networks*, vol. 9, no. 1, pp. 224–229, 1998.

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

[44] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *CoRR*, vol. abs/1406.1078, 2014.

[45] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*. Haifa, IS: ACM, 2010, pp. 807–814.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 770–778.

[47] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proceedings of icml*, vol. 30, no. 1. Atlanta, US: ACM, 2013, p. 3.

[48] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, vol. abs/1511.06434, 2015.

[49] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1725–1732.

[50] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1510–1517, 2018.

[51] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 4724–4733.

[52] O. M. Parkhi, A. Vedaldi, A. Zisserman *et al.*, "Deep face recognition." in *BMVC*, vol. 1, no. 3, 2015, p. 6.

[53] S. S. Farfade, M. J. Saberian, and L.-J. Li, "Multi-view face detection using deep convolutional neural networks," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. ACM, 2015, pp. 643–650.

[54] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5325–5334.

[55] D. Scherer, A. C. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Artificial Neural Networks - ICANN 2010 - 20th International Conference, Thessaloniki, Greece, September 15-18, 2010, Proceedings, Part III*, 2010, pp. 92–101.

[56] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 111–118.

[57] A. Géron, *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.", 2017.

[58] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, 2015, pp. 649–657.

[59] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[60] S. Poria, H. Peng, A. Hussain, N. Howard, and E. Cambria, "Ensemble application of convolutional neural networks and multiple kernel learning for multimodal sentiment analysis," *Neurocomputing*, vol. 261, pp. 217–230, 2017.

[61] J. Wehrmann, W. Becker, H. E. Cagnini, and R. C. Barros, "A character-based convolutional neural network for language-agnostic twitter sentiment analysis," in *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 2384–2391.

[62] S. Amiriparian, N. Cummins, S. Ottl, M. Gerczuk, and B. Schuller, "Sentiment analysis using image-based deep spectrum features," in *Proceedings of the 7th biannual Conference on Affective Computing and Intelligent Interaction (ACII 2017)*, San Antonio, TX, 2017, pp. 26–29.

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

[64] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *Proceedings of the 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, IEEE. Boston, US: IEEE, 2015, pp. 1–6.

[65] M. Valenti, A. Diment, G. Parascandolo, S. Squartini, and T. Virtanen, "Dcase 2016 acoustic scene classification using convolutional neural networks," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE 2016)*, 2016, pp. 95–99.

[66] T. Lidy and A. Schindler, "Cqt-based convolutional neural networks for audio scene classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, vol. 90. DCASE2016 Challenge, 2016, pp. 1032–1048.

[67] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Proceedings of INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association*. Makuhari, Chiba, Japan: ISCA, September 2010, pp. 1045–1048.

[68] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2013, pp. 6645–6649.

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

[70] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[71] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[72] S. Hochreiter and J. Schmidhuber, "Lstm can solve hard long time lag problems," in *Advances in Neural Information Processing Systems*, 1997, pp. 473–479.

[73] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Miami, FL: IEEE, June 2009, pp. 248–255.

[74] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[75] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.

[76] G. Hu, Y. Yang, D. Yi, J. Kittler, W. Christmas, S. Z. Li, and T. Hospedales, "When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition," in *Proceedings of the IEEE international conference on computer vision workshops*, 2015, pp. 142–150.

[77] S. Amiriparian, J. Pohjalainen, E. Marchi, S. Pugachevskiy, and B. Schuller, "Is deception emotional? an emotion-driven predictive approach," in *Proceedings of INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association*. San Francisco, CA: ISCA, September 2016, pp. 2011–2015.

[78] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1717–1724.

[79] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.

[80] J.-C. Chen, V. M. Patel, and R. Chellappa, "Unconstrained face verification using deep cnn features," in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*. IEEE, 2016, pp. 1–9.

[81] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *CoRR*, vol. abs/1411.7923, 2014.

[82] D. Marmanis, M. Datcu, T. Esch, and U. Stilla, "Deep learning earth observation classification using imagenet pretrained networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 1, pp. 105–109, 2016.

[83] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4534–4542.

[84] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler, "Deep learning for emotion recognition on small datasets using transfer learning," in *Proceedings of the 2015 ACM on international conference on multimodal interaction*. ACM, 2015, pp. 443–449.

[85] Y. Wei, Y. Zhao, C. Lu, S. Wei, L. Liu, Z. Zhu, and S. Yan, "Cross-modal retrieval with cnn visual features: A new baseline," *IEEE transactions on cybernetics*, vol. 47, no. 2, pp. 449–460, 2017.

[86] B. Kieffer, M. Babaie, S. Kalra, and H. R. Tizhoosh, "Convolutional neural networks for histopathology image classification: training vs. using pre-trained networks," in *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, 2017, pp. 1–6.

[87] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[88] S. Amiriparian, M. Gerczuk, S. Ottl, N. Cummins, S. Pugachevskiy, and B. Schuller, "Bag-of-deep-features: Noise-robust deep feature representations for audio analysis," in *Proceedings of the 31st International Joint Conference on Neural Networks (IJCNN)*. Rio de Janeiro, Brazil: IEEE, July 2018, pp. 2419–2425.

[89] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Advances in Neural Information Processing Systems*, Barcelona, Spain, 2016, pp. 892–900.

[90] E. Grinstein, N. Q. Duong, A. Ozerov, and P. Pérez, "Audio style transfer," in *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 586–590.

[91] D. Joshi, M. Merler, Q.-B. Nguyen, S. Hammer, J. Kent, J. R. Smith, and R. S. Feris, "Ibm high-five: Highlights from intelligent video engine," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 1249–1250.

[92] C. Hori, T. Hori, G. Wichern, J. Wang, T.-y. Lee, A. Cherian, and T. K. Marks, "Multimodal attention for fusion of audio and spatiotemporal features for video description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2528–2531.

[93] J. F. Gemmeke, D. P. W. 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 *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, March 2017, pp. 776–780.

[94] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "Cnn architectures for large-scale audio classification," in *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. New Orleans, LA: IEEE, 2017, pp. 131–135.

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

[96] ——, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.

[97] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Advances in neural information processing systems*, 2016, pp. 2172–2180.

[98] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[99] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *CoRR*, vol. abs/1701.07875, 2017.

[100] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," *CoRR*, vol. abs/1701.04862, 2017.

[101] R. Valle, W. Cai, and A. Doshi, "Tequilagan: How to easily identify gan samples," *CoRR*, vol. abs/1807.04919, 2018.

[102] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*. Helsinki, FI: ACM, 2008, pp. 1096–1103.

[103] 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.

[104] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1422–1432.

[105] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[106] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298–2304, November 2017.

[107] Z. Zuo, B. Shuai, G. Wang, X. Liu, X. Wang, B. Wang, and Y. Chen, "Convolutional recurrent neural networks: Learning spatial dependencies for image representation," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 18–26.

[108] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. Schuller, and S. Zafeiriou, "Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network," in *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, IEEE. Shanghai, CN: IEEE, 2016, pp. 5200–5204.

[109] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 5, pp. 927–939, May 2016.

[110] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2392–2396.

[111] B. Wu, A. Wan, X. Yue, and K. Keutzer, "Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1887–1893.

[112] J. Wehrmann, G. S. Sim
textasciitilde oes, R. C. Barros, and V. F. Cavalcante, "Adult content detection in videos with convolutional and recurrent neural networks," *Neurocomputing*, vol. 272, pp. 432–438, 2018.

[113] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Large-scale weakly supervised audio classification using gated convolutional neural network," in *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 121–125.

[114] M. Hyeongi, B. Joon, K. Bum-Jun, J. Shin-hyuk, J. Youngho, P. Young-cheol, and P. Sung-wook, "End-to-end crnn architectures for weakly supervised sound event detection," DCASE2018 Challenge, Tech. Rep., September 2018.

[115] W. Lim, S. Suh, and Y. Jeong, "Weakly labeled semi-supervised sound event detection using crnn with inception module," DCASE2018 Challenge, Tech. Rep., September 2018.

[116] H. Lim, J. Park, and Y. Han, "Rare sound event detection using 1D convolutional recurrent neural networks," DCASE2017 Challenge, Tech. Rep., September 2017.

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

[118] S. Adavanne, K. Drossos, E. Çakir, and T. Virtanen, "Stacked convolutional and recurrent neural networks for bird audio detection," in *Signal Processing Conference (EUSIPCO), 2017 25th European*. IEEE, 2017, pp. 1729–1733.

[119] E. Cakir, S. Adavanne, G. Parascandolo, K. Drossos, and T. Virtanen, "Convolutional recurrent neural networks for bird audio detection," in *Signal Processing Conference (EUSIPCO), 2017 25th European*. IEEE, 2017, pp. 1744–1748.

[120] T. Iqbal, Q. Kong, M. Plumbley, and W. Wang, "Stacked convolutional neural networks for general-purpose audio tagging," DCASE2018 Challenge, Tech. Rep., September 2018.

[121] X. Gao, S. Lin, and T. Y. Wong, "Automatic feature learning to grade nuclear cataracts based on deep learning," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 11, pp. 2693–2701, 2015.

[122] C. Qin, J. V. Hajnal, D. Rueckert, J. Schlemper, J. Caballero, and A. N. Price, "Convolutional recurrent neural networks for dynamic mr image reconstruction," *IEEE transactions on medical imaging*, 2018.

[123] T. Ma, C. Xiao, and F. Wang, "Health-atm: A deep architecture for multifaceted patient health record representation and risk prediction," in *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 2018, pp. 261–269.

[124] F. J. Ordó
textasciitilde nez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[125] D. Quang and X. Xie, "Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences," *Nucleic acids research*, vol. 44, no. 11, pp. e107–e107, 2016.

[126] X. Pan, P. Rijnbeek, J. Yan, and H.-B. Shen, "Prediction of rna-protein sequence and structure binding preferences using deep convolutional and recurrent neural networks," *BMC genomics*, vol. 19, no. 1, p. 511, 2018.

[127] S. Amiriparian, A. Baird, S. Julka, A. Alcorn, S. Ottl, S. Petrović, E. Ainger, N. Cummins, and B. Schuller, "Recognition of echolalic autistic child vocalisations utilising convolutional recurrent neural networks," in *Proceedings of INTERSPEECH 2018, 19th Annual Conference of the International Speech Communication Association*. Hyderabad, India: ISCA, September 2018, pp. 2334–2338.

[128] S. Amiriparian, S. Julka, N. Cummins, and B. Schuller, "Deep convolutional recurrent neural networks for rare sound event detection," in *Proceedings of 44. Jahrestagung für Akustik (DAGA)*, Munich, Germany, March 2018, pp. 1522–1525.

[129] S. C. Turaga, J. F. Murray, V. Jain, F. Roth, M. Helmstaedter, K. Briggman, W. Denk, and H. S. Seung, "Convolutional networks can learn to generate affinity graphs for image segmentation," *Neural Computation*, vol. 22, no. 2, pp. 511–538, 2010.

[130] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 609–616.

[131] N. Pinto, D. Doukhan, J. J. DiCarlo, and D. D. Cox, "A high-throughput screening approach to discovering good forms of biologically inspired visual representation," *PLoS Computational Biology*, vol. 5, no. 11, p. e1000579, 2009.

[132] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *2009 IEEE 12th International Conference on Computer Vision*, September 2009, pp. 2146–2153.

[133] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1299–1312, 2016.

[134] Y. Bar, I. Diamant, L. Wolf, S. Lieberman, E. Konen, and H. Greenspan, "Chest pathology detection using deep learning with non-medical training." in *ISBI*. Citeseer, 2015, pp. 294–297.

[135] S. Hoo-Chang, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, p. 1285, 2016.

[136] A. Baird, S. Amiriparian, N. Cummins, A. M. Alcorn, A. Batliner, S. Pugachevskiy, M. Freitag, M. Gerczuk, and B. Schuller, "Automatic classification of autistic child vocalisations: A novel database and results," in *Proceedings of INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association*. Stockholm, Sweden: ISCA, Augsust 2017, pp. 849–853.

[137] A. Baird, S. Amiriparian, A. Rynkiewicz, and B. Schuller, "Echolalic autism spectrum condition vocalisations: Brute-force and deep spectrum features," in *Proceedings of International Paediatric Conference (IPC 2018)*. Rzeszów, Poland: Polish Society of Social Medicine and Public Health, May 2018, 2 pages, to appear.

[138] S. Amiriparian, M. Schmitt, S. Hantke, V. Pandit, and B. Schuller, "Humans inside: Cooperative big multimedia data mining," in *Innovations in Big Data Mining and Embedded Knowledge: Domestic and Social Context Challenges*, ser. Intelligent Systems Reference Library (ISRL), A. Esposito, A. M. Esposito, and L. C. Jain, Eds. Springer, 2018, 25 pages, to appear.

[139] F. Ringeval, B. Schuller, M. Valstar, R. Cowie, H. Kaya, M. Schmitt, S. Amiriparian, N. Cummins, D. Lalanne, A. Michaud, E. Ciftçi, H. Güleç, A. A. Salah, and M. Pantic, "Avec 2018 workshop and challenge: Bipolar disorder and cross-cultural affect recognition," in *Proceedings of the 2018 on Audio/Visual Emotion Challenge and Workshop*, ser. AVEC'18. Seoul, Republic of Korea: ACM, 2018, pp. 3–13.

[140] F. J. Harris, "On the use of windows for harmonic analysis with the discrete fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, 1978.

[141] A. Nuttall, "Some windows with very good sidelobe behavior," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 1, pp. 84–91, 1981.

[142] M. Mauch and S. Dixon, "Simultaneous estimation of chords and musical context from audio," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1280–1289, 2010.

[143] 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.

[144] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Boston, MA, USA: IEEE, 2015, pp. 1–9.

[145] M. Schmitt and B. Schuller, "openXBOW – Introducing the Passau open-source crossmodal bag-of-words toolkit," *Journal of Machine Learning Research*, vol. 18, no. 96, pp. 1–5, October 2017.

[146] S. Pancoast and M. Akbacak, "Bag-of-audio-words approach for multimedia event classification," in *Proceedings of INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association*. Portland, OR: ISCA, September 2012, pp. 2105–2108.

[147] ——, "Softening quantization in bag-of-audio-words," in *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Florence, Italy: IEEE, May 2014, pp. 1370–1374.

[148] 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 *Proceedings of INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association*. San Francisco, CA: ISCA, September 2016, pp. 495–499.

[149] F. Pokorny, F. Graf, F. Pernkopf, and B. Schuller, "Detection of negative emotions in speech signals using bags-of-audio-words," in *Proceedings of the 1st International Workshop on Automatic Sentiment Analysis in the Wild (WASA 2015) held in conjunction with the 6th biannual Conference on Affective Computing and Intelligent Interaction (ACII 2015)*, AAAC. Xi'an, P. R. China: IEEE, September 2015, pp. 879–884.

[150] 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 *Proceedings of INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association*. Lyon, France: ISCA, August 2013, pp. 2929–2933.

[151] S. Amiriparian, S. Pugachevskiy, N. Cummins, S. Hantke, J. Pohjalainen, G. Keren, and B. Schuller, "CAST a database: Rapid targeted large-scale big data acquisition via small-world modelling of social media platforms," in *Proceedings of the 7th biannual Conference on Affective Computing and Intelligent Interaction (ACII 2017)*, San Antonio, TX, 2017, pp. 340–345.

[152] S. Amiriparian, M. Freitag, N. Cummins, M. Gerzcuk, S. Pugachevskiy, and B. W. Schuller, "A fusion of deep convolutional generative adversarial networks and sequence to sequence autoencoders for acoustic scene classification," in *Proceedings of 26th European Signal Processing Conference (EUSIPCO)*, EURASIP. Rome, Italy: IEEE, September 2018, pp. 982–986.

[153] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," in *3rd International Conference on Learning Representations*, San Diego, US, 2015.

[154] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved techniques for training GANs," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 2234–2242.

[155] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[156] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[157] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3104–3112.

[158] M. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, "Multi-task sequence to sequence learning," *CoRR*, vol. abs/1511.06114, 2015.

[159] A. M. Dai and Q. V. Le, "Semi-supervised sequence learning," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 3079–3087.

[160] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Józefowicz, and S. Bengio, "Generating sentences from a continuous space," *CoRR*, vol. abs/1511.06349, 2015.

[161] M. Jang, S. Seo, and P. Kang, "Recurrent neural network-based semantic variational autoencoder for sequence-to-sequence learning," *CoRR*, vol. abs/1802.03238, 2018.

[162] F. Weninger, S. Watanabe, Y. Tachioka, and B. Schuller, "Deep recurrent denoising auto-encoder and blind de-reverberation for reverberated speech recognition," in *International Conference on Acoustics, Speech and Signal Processing*. Florence, IT: IEEE, 2014, pp. 4623–4627.

[163] S. Amiriparian, M. Schmitt, N. Cummins, K. Qian, F. Dong, and B. Schuller, "Deep unsupervised representation learning for abnormal heart sound classification," in *Proceedings of the 40th Annual International Conference of the IEEE Engineering in Medicine & Biology Society, EMBC 2018*, IEEE. Honolulu, HI: IEEE, July 2018, pp. 4776–4779.

[164] B. W. Schuller, S. Steidl, A. Batliner, P. B. Marschik, H. Baumeister, F. Dong, S. Hantke, F. Pokorny, E.-M. Rathner, K. D. Bartl-Pokorny, C. Einspieler, D. Zhang, A. Baird, S. Amiriparian, K. Qian, Z. Ren, M. Schmitt, P. Tzirakis, and S. Zafeiriou, "The interspeech 2018 computational paralinguistics challenge: Atypical & self-assessed affect, crying & heart beats," in *Proceedings of INTERSPEECH 2018, 19th Annual Conference of the International Speech Communication Association*. Hyderabad, India: ISCA, September 2018, pp. 122–126.

[165] S. Amiriparian, M. Freitag, N. Cummins, and B. Schuller, "Feature selection in multimodal continuous emotion prediction," in *Proceedings of the 2nd International Workshop on Automatic Sentiment Analysis in the Wild (WASA 2017) held in conjunction with the 7th biannual Conference on Affective Computing and Intelligent Interaction (ACII 2017)*, AAAC. San Antonio, TX: IEEE, October 2017, pp. 30–37.

[166] R. Bellman, *Dynamic programming*. Courier Corporation, 2013.

[167] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.

[168] H. Kaya, F. Çilli, and A. A. Salah, "Ensemble cca for continuous emotion prediction," in *Proceedings of AVEC'14*. Orlando, FL, US: ACM, 2014, pp. 19–26.

[169] M. A. Hall, "Correlation-based feature selection for discrete and numeric class machine learning," in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 359–366.

[170] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1, pp. 273–324, 1997.

[171] 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.

[172] S. Gu, R. Cheng, and Y. Jin, "Feature selection for high-dimensional classification using a competitive swarm optimizer," *Soft Computing*, pp. 1–12, 2016.

[173] M. Freitag, S. Amiriparian, N. Cummins, M. Gerczuk, and B. Schuller, "An 'end-to-evolution' hybrid approach for snore sound classification," in *Proceedings of INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association*. Stockholm, Sweden: ISCA, August 2017, pp. 3507–3511.

[174] Y. Hoshen, R. J. Weiss, and K. W. Wilson, "Speech acoustic modeling from raw multichannel waveforms," in *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4624–4628.

[175] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform cldnns," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

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

[177] B. Logan *et al.*, "Mel frequency cepstral coefficients for music modeling." in *ISMIR*, vol. 270, 2000, pp. 1–11.

[178] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *24th European Signal Processing Conference (EUSIPCO 2016)*. Budapest, Hungary: IEEE, Aug 2016, pp. 1128–1132.

[179] 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 *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 IEEE AASP Challenge Workshop (DCASE 2016), satellite to EUSIPCO 2016*, EUSIPCO. Budapest, Hungary: IEEE, September 2016, pp. 1–5.

[180] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, October 2015.

[181] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer, "CP-JKU submissions for DCASE-2016: A hybrid approach using binaural i-vectors and deep convolutional neural networks," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2016.

[182] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations*, Banff, CA, 2014.

[183] P. K. Atrey, N. C. Maddage, and M. S. Kankanhalli, "Audio based event detection for multimedia surveillance," in *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5. IEEE, 2006, pp. V–V.

[184] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in *Signal Processing Conference, 2010 18th European*. IEEE, 2010, pp. 1267–1271.

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

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

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

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

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

[190] K. Collins, *Playing with Sound : A Theory of Interacting with Sound and Music in Video Games*, 1st ed. Cambridge, MA, USA: The MIT Press, 2013.

[191] ——, "Game sound," *An introduction to the history, theory, and practice of video game music and sound design, Cambridge*, 2008.

[192] S. Egenfeldt-Nielsen, J. H. Smith, and S. P. Tosca, *Understanding video games: The essential introduction*, 3rd ed. New York, NY, USA: Routledge, 2016.

[193] M. Frutos-Pascual and B. G. Zapirain, "Review of the use of AI techniques in serious games: Decision making and machine learning," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 9, no. 2, pp. 133–152, June 2017.

[194] M. C. Gombolay, R. Jensen, and S. H. Son, "Machine learning techniques for analyzing training behavior in serious gaming," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. PP, no. 99, pp. 1–1, 2017.

[195] Steinkuehler, Constance, "Parenting and video games," *Journal of Adolescent & Adult Literacy*, vol. 59, no. 4, pp. 357–361, 2016.

[196] I. G. Initiative, "Ethically aligned design," https://ethicsinaction.ieee.org, 2018, accessed: 1-11-2018.

[197] S. Amiriparian, N. Cummins, M. Gerczuk, S. Pugachevskiy, S. Ottl, and B. Schuller, ""are you playing a shooter again?!" deep representation learning for audio-based video game genre recognition," *IEEE Transactions on Games*, vol. 11, 2018.

[198] B. Schuller, S. Steidl, and A. Batliner, "The interspeech 2009 emotion challenge," in *Proceedings of INTERSPEECH 2009, 10th Annual Conference of the International Speech Communication Association*. Brighton, UK: ISCA, September 2009, pp. 312–315.

[199] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Müller, and S. Narayanan, "The interspeech 2010 paralinguistic challenge," in *Proceedings of INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association*. Makuhari, Japan: ISCA, September 2010, pp. 2794–2797.

[200] A. Austin, E. Moore, U. Gupta *et al.*, "Characterization of movie genre based on music score," in *Proceedings of the 2010 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Dallas, TX, USA: IEEE, Mar 2010, pp. 421–424.

[201] A. Rosner, F. Weninger, B. Schuller *et al.*, "Influence of low-level features extracted from rhythmic and harmonic sections on music genre classification," in *Man-Machine Interaction 3*, ser. Advances in Intelligent Systems and Computing (AISC), A. Gruca, T. Czachórski, and S. Kozielski, Eds. Springer, 2014, vol. 242, pp. 467–473.

[202] A. Rosner, B. Schuller, and B. Kostek, "Classification of music genres based on music separation into harmonic and drum components," *Archives of Acoustics*, vol. 39, no. 4, pp. 629–638, 2015.

[203] F. Eyben, *Real-time Speech and Music Classification by Large Audio Feature Space Extraction*, ser. Springer Theses. Springer International Publishing, 2015.

[204] S. Panwar, A. Das, M. Roopaei, and P. Rad, "A deep learning approach for mapping music genres," in *Proceedings of the 12th Conference on System of Systems Engineering Conference (SoSE 2017)*. IEEE, 2017, pp. 1–5.

[205] J. Salamon and J. P. Bello, "Unsupervised feature learning for urban sound classification," in *International Conference on Acoustics, Speech and Signal Processing*. Brisbane, AU: IEEE, 2015, pp. 171–175.

[206] B. McFee, M. McVicar, O. Nieto, S. Balke, C. Thome, D. Liang, E. Battenberg, J. Moore, R. Bittner, R. Yamamoto, D. Ellis, F.-R. Stoter, D. Repetto, S. Waloschek,

C. Carr, S. Kranzler, K. Choi, P. Viktorin, J. F. Santos, A. Holovaty, W. Pimenta, and H. Lee, "librosa 0.5.0," February 2017.

[207] J. Peat and B. Barton, *Medical statistics: A guide to data analysis and critical appraisal*. John Wiley & Sons, 2008.

[208] D. Öztuna, A. H. Elhan, and E. Tüccar, "Investigation of four different normality tests in terms of type 1 error rate and power under different distributions," *Turkish Journal of Medical Sciences*, vol. 36, no. 3, pp. 171–176, 2006.

[209] S. Hantke, F. Eyben, T. Appel *et al.*, "iHEARu-PLAY: introducing a game for crowdsourced data collection for affective computing," in *Proceedings of the 1st International Workshop on Automatic Sentiment Analysis in the Wild (WASA 2015) held conjunction with 6th biannual Conference on Affective Computing and Intelligent Interaction (ACII 2015)*, AAAC. Xi'an, P. R. China: IEEE, September 2015, pp. 891–897.

[210] A. Marchand and T. Hennig-Thurau, "Value creation in the video game industry: Industry economics, consumer benefits, and research opportunities," *Journal of Interactive Marketing*, vol. 27, no. 3, pp. 141–157, 2013.

[211] M. Wöllmer, F. Weninger, T. Knaup, B. Schuller, C. Sun, K. Sagae, and L.-P. Morency, "Youtube movie reviews: Sentiment analysis in an audio-visual context," *IEEE Intelligent Systems*, vol. 28, no. 3, pp. 46–53, 2013.

[212] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of machine learning research*, vol. 9, no. Aug, pp. 1871–1874, 2008.

[213] T. Joachims, "Training linear svms in linear time," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2006, pp. 217–226.

[214] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, vol. 4, 2017, p. 12.

[215] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," *CoRR*, vol. abs/1609.08675, 2016.

[216] 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, February 2018.

[217] K. R. Scherer, "What are emotions? and how can they be measured?" *Social science information*, vol. 44, no. 4, pp. 695–729, 2005.

[218] K. L. Burns and E. G. Beier, "Significance of vocal and visual channels in the decoding of emotional meaning," *Journal of Communication*, vol. 23, no. 1, p. 118, 1973.

[219] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," *IEEE Intelligent Systems*, vol. 28, no. 2, pp. 15–21, 2013.

[220] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[221] 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.

[222] C. Ittichaichareon, S. Suksri, and T. Yingthawornsuk, "Speech recognition using mfcc," in *International Conference on Computer Graphics, Simulation and Modeling (ICGSM'2012) July*, 2012, pp. 28–29.

[223] N. Dave, "Feature extraction methods lpc, plp and mfcc in speech recognition," *International journal for advance research in engineering and technology*, vol. 1, no. 6, pp. 1–4, 2013.

[224] D. Mozaffarian, E. J. Benjamin, A. Go, D. K. Arnett, M. J. Blaha *et al.*, "Executive summary: Heart disease and stroke statistics-2016 update: A report from the american heart association," *Circulation*, vol. 133, no. 4, pp. 447–454, 2016.

[225] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. S. Kwak, "The internet of things for health care: A comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.

[226] 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 *Proceedings of INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association*. Lyon, France: ISCA, August 2013, pp. 148–152.

[227] M. Schmitt and B. 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.

[228] V. Maknickas and A. Maknickas, "Recognition of normal-abnormal phonocardiographic signals using deep convolutional neural networks and mel-frequency spectral coefficients," *Physiological Measurement*, vol. 38, no. 8, p. 1671, 2017.

[229] B. Schuller, S. Steidl, A. Batliner, J. Hirschberg, J. K. Burgoon, A. Baird, A. Elkins, Y. Zhang, E. Coutinho, and K. Evanini, "The interspeech 2016 computational paralinguistics challenge: Deception, sincerity & native language," in *Proceedings of INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association*. San Francisco, CA: ISCA, September 2016, pp. 2001–2005.

[230] F. Weninger, F. Eyben, B. W. Schuller, M. Mortillaro, and K. R. Scherer, "On the acoustics of emotion in audio: What speech, music and sound have in common," *Frontiers in Psychology, section Emotion Science, Special Issue on Expression of emotion in music and vocal communication*, vol. 4, no. Article ID 292, pp. 1–12, May 2013.

[231] J. C. Platt, "12 fast training of support vector machines using sequential minimal optimization," *Advances in kernel methods*, pp. 185–208, 1999.

[232] C. Liu, D. Springer, Q. Li, B. Moody, R. A. Juan *et al.*, "An open access database for the evaluation of heart sound algorithms," *Physiological Measurement*, vol. 37, no. 12, pp. 2181–2213, 2016.

[233] M. S. Aldrich, *Sleep medicine*. Oxford University Press, 1999.

[234] I. Fietze, T. Penzel, A. Alonderis, F. Barbe, M. Bonsignore, P. Calverly, W. De Backer, K. Diefenbach, V. Donic, M. Eijsvogel *et al.*, "Management of obstructive sleep apnea in europe," *Sleep medicine*, vol. 12, no. 2, pp. 190–197, 2011.

[235] T. Young, L. Evans, L. Finn, M. Palta *et al.*, "Estimation of the clinically diagnosed proportion of sleep apnea syndrome in middle-aged men and women," *Sleep*, vol. 20, no. 9, pp. 705–706, 1997.

[236] P. Jennum and R. L. Riha, "Epidemiology of sleep apnoea/hypopnoea syndrome and sleep-disordered breathing," *European Respiratory Journal*, vol. 33, no. 4, pp. 907–914, 2009.

[237] J. C. Lam, S. Sharma, B. Lam *et al.*, "Obstructive sleep apnoea: definitions, epidemiology & natural history," *Indian Journal of Medical Research*, vol. 131, no. 2, p. 165, 2010.

[238] A. S. Shamsuzzaman, B. J. Gersh, and V. K. Somers, "Obstructive sleep apnea: implications for cardiac and vascular disease," *Jama*, vol. 290, no. 14, pp. 1906–1914, 2003.

[239] O. Parra, A. Arboix, J. Montserrat, L. Quinto, S. Bechich, and L. Garcia-Eroles, "Sleep-related breathing disorders: impact on mortality of cerebrovascular disease," *European Respiratory Journal*, vol. 24, no. 2, pp. 267–272, 2004.

[240] C. Croft and M. Pringle, "Sleep nasendoscopy: a technique of assessment in snoring and obstructive sleep apnoea," *Clinical Otolaryngology*, vol. 16, no. 5, pp. 504–509, 1991.

[241] P. Hill, B. Lee, J. Osborne, and E. Osman, "Palatal snoring identified by acoustic crest factor analysis," *Physiological measurement*, vol. 20, no. 2, p. 167, 1999.

[242] 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.

[243] 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*, 2016.

[244] B. Schuller, A. Batliner, S. Steidl, F. Schiel, and J. Krajewski, "The interspeech 2011 speaker state challenge," in *Proceedings of INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association*. Florence, Italy: ISCA, August 2011, pp. 3201–3204.

[245] B. Schuller, J.-G. Ganascia, and L. Devillers, "Multimodal sentiment analysis in the wild: Ethical considerations on data collection, annotation, and exploitation," in *Proceedings of International Workshop on ETHics In Corpus Collection, Annotation and Application (ETHI-CA^2), satellite of the Language Resources and Evaluation Conference (LREC)*, Portoroz, Slovenia, 2016, pp. 29–34.

[246] W. Samek, T. Wiegand, and K. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," *CoRR*, vol. abs/1708.08296, 2017.

[247] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2016, pp. 1135–1144.

[248] Z. C. Lipton, "The mythos of model interpretability," *CoRR*, vol. abs/1606.03490, 2016.

[249] G. Marcus, "Deep learning: A critical appraisal," *CoRR*, vol. abs/1801.00631, 2018.

[250] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 2010, pp. 249–256.

[251] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[252] B. M. Lake and M. Baroni, "Still not systematic after all these years: On the compositional skills of sequence-to-sequence recurrent networks," *CoRR*, vol. abs/1711.00350, 2017.

[253] A. Graves, "Generating sequences with recurrent neural networks," *CoRR*, vol. abs/1308.0850, 2013.

[254] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014.

[255] B. Schuller, *Intelligent Audio Analysis*, ser. Signals and Communication Technology. Springer, 2013, 350 pages.

[256] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE transactions on audio, speech, and language processing*, vol. 15, no. 3, pp. 1066–1074, 2007.

[257] T. Heittola, A. Mesaros, T. Virtanen, and A. Eronen, "Sound event detection in multisource environments using source separation," in *Proceedings of Machine Listening in Multisource Environments (CHiME)*, Florence, Italy, 2011, pp. 36–40.

[258] T. Barker and T. Virtanen, "Non-negative tensor factorisation of modulation spectrograms for monaural sound source separation," in *Proceedings of INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association*. Lyon, France: ISCA, Augsust 2013, pp. 827–831.

[259] J. Nikunen, A. Diment, and T. Virtanen, "Separation of moving sound sources using multichannel nmf and acoustic tracking," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 26, no. 2, pp. 281–295, 2018.

[260] H. Bredin, "Tristounet: triplet loss for speaker turn embedding," in *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5430–5434.

[261] G. Wisniewksi, H. Bredin, G. Gelly, and C. Barras, "Combining speaker turn embedding and incremental structure prediction for low-latency speaker diarization," in *Proceedings of INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association*. Stockholm, Sweden: ISCA, Augsust 2017, pp. 3582–3586.

[262] D. A. Reynolds and P. Torres-Carrasquillo, "Approaches and applications of audio diarization," in *Proceedings of the 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5. IEEE, 2005, pp. 953–956.

[263] B. Marr, *Big Data: Using SMART big data, analytics and metrics to make better decisions and improve performance*. John Wiley & Sons, 2015.

[264] Y. Wang, L. Kung, and T. A. Byrd, "Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations," *Technological Forecasting and Social Change*, vol. 126, pp. 3–13, 2018.

[265] V. Pandit, S. Amiriparian, M. Schmitt, K. Qian, J. Guo, S. Matsuoka, and B. Schuller, "Big data multimedia mining: Feature extraction facing volume, velocity, and variety," in *Big Data Analytics for Large-Scale Multimedia Search*, S. Vrochidis, B. Huet, E. Chang, and I. Kompatsiaris, Eds. Wiley, 2017.

[266] W. Han, E. Coutinho, H. Ruan, H. Li, B. Schuller, X. Yu, and X. Zhu, "Semi-supervised active learning for sound classification in hybrid learning environments," *PLoS ONE*, vol. 11, no. 9, 2016, 23 pages.

[267] 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 (TASLP)*, vol. 23, no. 1, pp. 115–126, 2015.

[268] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th International Conference on Machine Learning*. Bellevue, US: ACM, 2011, pp. 689–696.

[269] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," in *Advances in neural information processing systems*, 2012, pp. 2222–2230.

[270] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.