Tampere University

Ville-Veikko Eklund

# DATA AUGMENTATION TECHNIQUES FOR ROBUST AUDIO ANALYSIS

Faculty of Information Technology and Communication Sciences
Master of Science Thesis
September 2019

# ABSTRACT

Ville-Veikko Eklund: Data Augmentation Techniques for Robust Audio Analysis
Master of Science Thesis
Tampere University
Degree Programme in Electrical Engineering, MSc (Tech)
September 2019

---

Having large amounts of training data is necessary for the ever more popular neural networks to perform reliably. Data augmentation, i.e. the act of creating additional training data by performing label-preserving transformations for existing training data, is an efficient solution for this problem. While increasing the amount of data, introducing variations to the data via the transformations also has the power to make machine learning models more robust in real life conditions with noisy environments and mismatches between the training and test data.

In this thesis, data augmentation techniques in audio analysis are reviewed, and a tool for audio data augmentation (TADA) is presented. TADA is capable of performing three audio data augmentation techniques, which are convolution with mobile device microphone impulse responses, convolution with room impulse responses, and addition of background noises. TADA is evaluated by using it in a pronunciation error classification task, where typical pronunciation errors of Finnish people uttering English words are classified. All the techniques are tested first individually and then also in combination.

The experiments are executed with both original and augmented data. In all experiments, using TADA improves the performance of the classifier when compared to training with only original data. Robustness against unseen devices and rooms also improves. Additional gain from performing combined augmentation starts to saturate only after augmenting the training data to 30 times the original amount. Based on the positive impact of TADA for the classification task, it is found that data augmentation with convolutional and additive noises is an effective combination for increasing robustness against environmental distortions and channel effects.

Keywords: data augmentation, audio analysis, robust classification, supervised learning, additive noise, impulse response

# TIIVISTELMÄ

Ville-Veikko Eklund: Aineiston täydennysmenetelmät robustia äänen analyysiä varten
Diplomityö
Tampereen yliopisto
Sähkötekniikan DI-tutkinto-ohjelma
Syyskuu 2019

---

Viime aikoina nopeasti yleistyneiden neuroverkkojen opettamiseksi tarvitaan suuria määriä dataa, jotta niistä saadaan luotettavia. Aineiston täydennys, eli lisäaineiston luominen suorittamalla luokkatunnuksen säilyttäviä muunnoksia olemassa olevalle aineistolle, on tehokas ratkaisu kyseiseen ongelmaan. Aineiston kasvattamisen lisäksi vaihteluiden lisääminen opetusdataan voi tehdä koneoppimismalleista robusteja kohinaista, todellista dataa kohtaan.

Tässä työssä käydään läpi äänen analyysissä käytettäviä aineiston täydennysmenetelmiä ja esitellään aineiston lisäämistä varten kehitetty täydennystyökalu. Työkaluun kehitetyt kolme erillistä aineiston täydennysmenetelmää ovat konvoluutio mobiililaitteiden mikrofonien impulssivasteiden kanssa, konvoluutio huoneimpulssivasteiden kanssa sekä taustakohinan lisäys. Työkalua testataan käyttämällä sitä lausumisvirheluokittelutehtävässä, jossa tarkoituksena on luokitella tyypillisiä suomalaisten tekemiä lausumisvirheitä englanninkielisissä sanoissa. Kaikki implementoidut menetelmät testataan aluksi erikseen ja lopuksi yhdessä.

Testit suoritetaan käyttämällä sekä alkuperäistä että täydennettyä testidataa. Kaikissa testeissä työkalua käyttämällä saadaan kasvatettua luokittelijan tarkkuutta verrattuna alkuperäisellä datalla opetettuun luokittelijaan. Robustius uusia mobiililaitteita ja huoneita kohtaan myös paranee. Tarkkuuden kasvu yhdistetyssä testissä saturoituu, kun opetusdata on täydennetty 30-kertaiseksi. Työkalun positiivisen vaikutuksen perusteella aineiston täydennys konvoluutioilla ja lisätyllä kohinalla osoittautuu tehokkaaksi menetelmäksi robustiuden lisäämiseksi ympäristön ja tallennusvälineiden aiheuttamia häiriöitä kohtaan.

Avainsanat: aineiston täydennys, äänen analyysi, robusti luokittelu, ohjattu oppiminen, lisätty kohina, impulssivaste

# PREFACE

This thesis was written during the spring and summer of 2019 at the former Laboratory of Signal Processing at Tampere University. The data for the thesis was collected during 2018.

I would like to thank the examiners of the thesis, Tuomas Virtanen and Aleksandr Diment, for their excellent guidance in the process, and Aleksandr also for his extraordinary supervision and invaluable advice. I am grateful for the opportunity to work in the Audio Research Group and for all the help I received from the members of the group. I would like to express my gratitude for CSC–IT Center for Science, Finland for providing the needed computing resources. Finally, I wish to thank my family for supporting me during this process.

Tampere, 30th September 2019

Ville-Veikko Eklund

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF SYMBOLS AND ABBREVIATIONS

| | |
|---|---|
| ASR | automatic speech recognition |
| CV | cross-validation |
| $\delta(t)$ | Dirac delta function |
| ESS | exponential sine sweep |
| FFT | fast Fourier transform |
| $H(\omega)$ | frequency response |
| $h(t)$ | impulse response |
| IR | impulse response |
| LSTM | long short-term memory |
| LTI system | linear time-invariant system |
| MFCC | mel-frequency cepstral coefficient |
| MIR | music information retrieval |
| MLS | maximum length sequence |
| RIR | room impulse response |
| RNN | recurrent neural network |
| SNR | signal-to-noise ratio |
| TADA | tool for audio data augmentation |
| TUT | Tampere University of Technology |
| VTLP | vocal tract length perturbation |
| WER | word error rate |

# 1 INTRODUCTION

The quick development of machine learning methods, and lately especially neural networks, has lead to an increasing need of large amounts of data. While collecting large datasets is a tedious and time-consuming task, the quality of data also has a great impact on the performance of a model. Machine learning models are expected to perform well on realistic and not only on laboratory quality data, which further increases the amount of resources needed for data collection. Obtaining realistic data becomes even more essential, when machine learning is being integrated with a growing rate into smartphones and other devices. These devices typically operate on data, which contains highly varying levels of noise and other disturbances.

## 1.1 Data augmentation

The ability of a machine learning model to cope with noise and distortions, i.e. *robustness*, can be improved with a number of methods, one of which is *data augmentation*. In data augmentation, existing data is altered for example by adding noise or by filtering it. The altered data is then added to the original training set, and this resulting augmented training set is used to train a machine learning model. A common example of image data augmentation is rotation. A human can easily recognise a rotated image to contain the same content as a non-rotated image, but for a machine learning model rotation is not necessarily a trivial concept. The model trained with augmented data is expected to be less susceptible for distortions and therefore more robust because the model has learned to ignore unimportant details.

Data augmentation can also be thought of as artificial data collection, since it increases the amount of data without the actual data collection process. Therefore, at the same time it is capable of reducing the considerable effort of labeling new data and increasing the variability of distortions in the data needed for making robust models.

## 1.2 Objectives

In this thesis, techniques to improve the robustness of machine learning models to environmental noise and channel effects are studied. The thesis focuses on audio data, and therefore only audio analysis tasks are covered. The main focus is on data augmenta-

tion techniques, and all the common techniques are studied in detail in the background section. Because impulse responses are tightly related to audio data augmentation and their measurement is relevant for the implementation part of the thesis, impulse response measurement techniques are also reviewed.

The main objective of the thesis is to create a data augmentation tool suitable for use in audio analysis tasks with a focus on data recorded with mobile devices. The tool for audio data augmentation (TADA) performs noise addition and convolutions with room and mobile device microphone impulse responses. With these functionalities it is possible to simulate effects of rooms and devices with a variable amount of background noise and therefore modify audio samples to have the characteristics of having been recorded in different places with different recording devices.

The applicability of the tool for audio analysis is evaluated by experiments with a neural network model designed for pronunciation error classification. There, the task is to classify utterances based on the presence of specific kinds of pronunciation errors. Such classifiers can be used in language teaching systems, where the goal is to improve pronunciation skills of language students. In this work, the classification was binary, i.e. there was only one error class, and it concentrated only on a specific phoneme of a word at a time.

## 1.3 Implementation

The implementation starts with the collection and selection of supporting datasets to be used with TADA. To implement the noise addition functionality for TADA for increasing robustness against additive environmental distortions, background noise samples are needed. Acoustic scenes, which are environments characterised by a typical audio background, are selected as the source of background noise because a decent number of good quality acoustic scene datasets is publicly available. The datasets are reviewed and the selection of datasets to be used is motivated.

For the convolution functionality aiming at increasing robustness against channel effects, all the impulse responses are measured instead of using ready datasets. Mobile device microphone impulse responses are not publicly available, so it is necessary to measure them. Although there are some room impulse response datasets available, measuring also them allows to better control the number of responses and measurement points.

Once the datasets are collected, the tool is implemented with Python as a class with a simple interface consisting of methods for the three augmentation techniques. To make it more straightforward to perform combinations of the three techniques, a method for stacking them on top of each other is prepared. In addition, the tool will partition the data used for augmentation to enable also test-time augmentation.

## 1.4 Organisation of the thesis

Chapter 2 begins with an introduction to supervised classification and audio analysis followed by causes of distortions in data and robust classification. Existing audio data augmentation techniques are reviewed and theory related to impulse responses and their measurement techniques is explained.

In Chapter 3, the proposed data augmentation tool TADA and the selected data augmentation techniques and their implementation are introduced. Specifications of the conducted impulse response measurements are also reported. TADA is then evaluated in Chapter 4 by incorporating it into a pronunciation error classifier and by testing the classifier in different augmentation scenarios. Finally, based on the results of evaluation, conclusions are drawn and further design ideas for TADA are discussed in Chapter 5.

# 2 BACKGROUND

In this chapter, supervised classification is briefly explained, fields of audio analysis are presented and the use of data augmentation in machine learning is motivated. In addition, existing audio data augmentation techniques and datasets suitable for augmentation are shown. Finally, impulse response theory and measurement techniques are covered.

## 2.1 Supervised classification

Supervised learning [45] is an area of pattern recognition, where functions for mapping objects to outputs are learned from examples of input-output pairs. Supervised learning is one of the three learning scenarios in pattern recognition with the other two being unsupervised learning and semi-supervised learning. In supervised learning, there are outputs or ground truths available for a set of objects called a training set, which is used to train a model. In unsupervised learning or clustering, the task is to group objects based only on their features without prior information of output values. The third major learning setting, semi-supervised learning, is a combination of both supervised and unsupervised learning, where samples with ground truths are used together with feature information from unlabeled data.

Supervised learning can further be divided into supervised classification and supervised regression. In classification, the goal is to predict class labels for unlabeled objects in a test set. These class labels are predefined based on the objects in a training set. In regression, continuous values are predicted instead of class labels. Steps of creating and evaluating a classifier in a supervised learning scenario are depicted in Figure 2.1.

### 2.1.1 Training and evaluation of a classifier

Supervised classification includes the following steps: data collection, data preprocessing, feature extraction, training, and evaluation. Data collection consists of selecting suitable existing datasets for the task or optionally recording the material and annotating it. In preprocessing, the data is prepared for feature extraction and it may include for example segmenting the audio into frames. Feature extraction aims to reduce the dimensionality of data and discard redundant information that could potentially make the learning task more difficult. In training, the data is fed to the classifier to construct a model of the

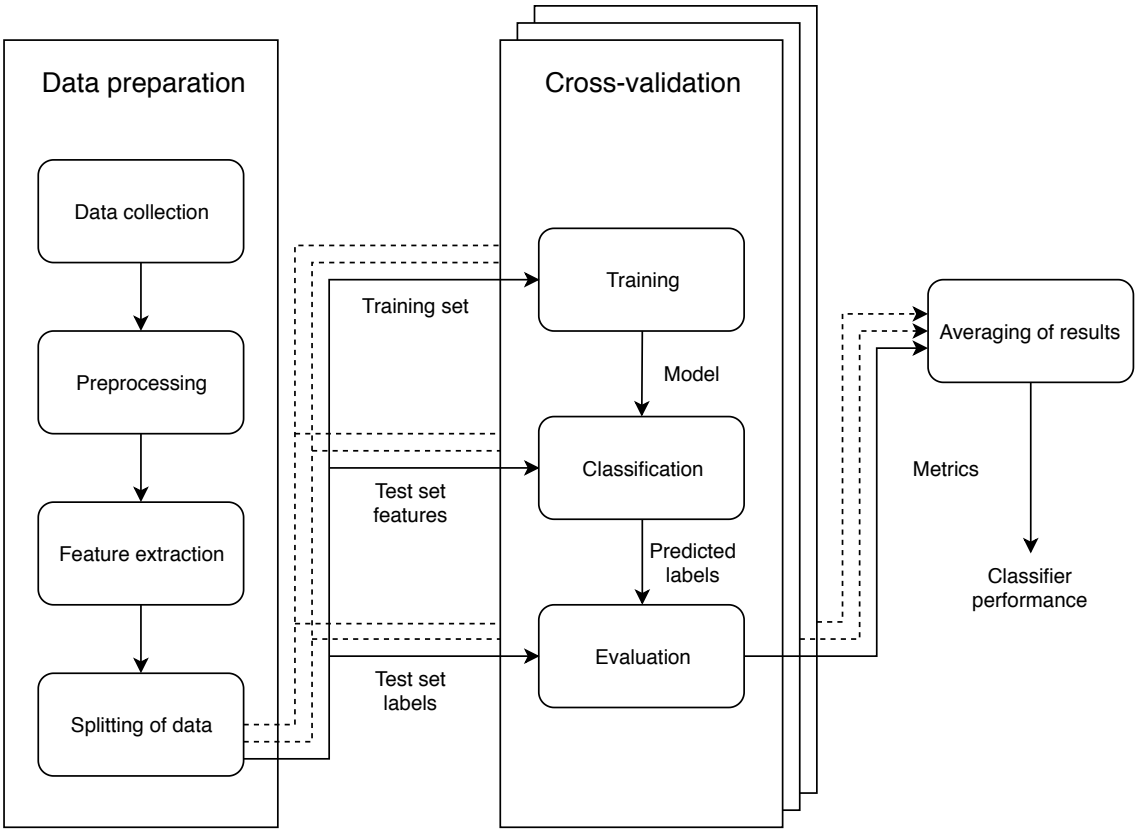


***Figure 2.1.*** *A supervised classification workflow.*

function between the input and the output. The type of data and the task may affect the selection of the classification method. For example, when using neural networks, recurrent neural networks (RNN) have been preferred with text data in natural language processing, and convolutional neural networks with image data.

To evaluate the goodness of a model, a set of objects called a test set is put aside before the training stage and left out from the training of the model. Once training is complete, the model is used to predict outputs for the objects in the test set and the selected metric determines how well the model has learned the desired mapping function. This validation technique is called hold-out, but there are also alternative techniques such as resubstitution, cross-validation and leave-one-out [53].

In resubstitution, the same data is used to train and test the model, which may result in overfitting and overly optimistic results. Overfitting means that the model learns all the little details in the training data and therefore achieves high accuracies when tested against the same data. However, the model does not generalise to other data anymore resulting in worse overall performance.

Because the performance of a model for a single test set is dependent on the split of data into training and test sets, cross-validation (CV) is usually performed. In cross-validation, the general idea is to split the data multiple times into training and test sets, and to train and measure the accuracy or some other performance metric of a model for each of the splits. Finally, the results for all splits are gathered and averaged to get

a more reliable measure of the performance of the learning method. If the data is split into $k$ non-overlapping subsets and each of the subsets is used once as a test set while the rest of the data is used for training, the procedure is called $k$-fold cross-validation. Another variation of cross-validation is Monte Carlo cross-validation, where the splits are done randomly.

Leave-one-out is a special case of $k$-fold cross-validation, where $k$ is equal to the total number of samples. In leave-one-out, the test set therefore consists of only one sample at a time while others are used for training. Although leave-one-out is a suitable method for a small amount of data, it is a very exhaustive and computationally heavy operation when compared to the other options.

### 2.1.2 Examples of audio analysis tasks

Audio analysis, which focuses on the extraction of information from audio, offers a variety of tasks suitable for supervised classification. The emphasis in these tasks is on different kinds of sounds, such as speech, music, and environmental sounds.

In automatic speech recognition (ASR) [57], the goal is to train systems to be able to recognize speech and transcribe it into text. ASR has been an active research area already for over half a century, and the applications include speech-to-speech translators, personal digital assistants, and living room interaction systems. The widely used audio features, mel-frequency cepstral coefficients (MFCCs), were originally designed for speech-related problems [33]. MFCCs are based on the mel scale [51], which corresponds to the perception of pitch by humans unlike a linear scale. Besides speech recognition, source separation and speech enhancement are active topics in the field. Signal enhancement generally is also discussed as one of the techniques used in robust classification in Section 2.3.2.

Music information retrieval (MIR) [41] concentrates on topics such as the recognition of instruments and genres, and automatic music transcription. Application possibilities for MIR include music recommendation systems, automatic music generators, and separation of individual instrument tracks from songs.

Sound event classification [54, Chapter 1] focuses on the classification of sound events, which are typically sounds made by animals, machines or natural phenomena. A closely related task is sound event detection, where the times of occurrences of possibly overlapping sound events are being detected. Apart from individual sound events, in acoustic scene classification the sound environments or the backgrounds consisting of a multitude of sound sources are being classified. Applications for sound event detection are for example smart home monitoring for security purposes, animal population monitoring, and context-based indexing in multimedia databases.

## 2.2 Environmental distortions

When a sound travels from its source to a listener or a recording microphone, the surrounding environment distorts the acoustic signal in a number of ways. These distortions can be divided into additive and convolutional noises [2] following the time-domain model illustrated in Figure 2.2.

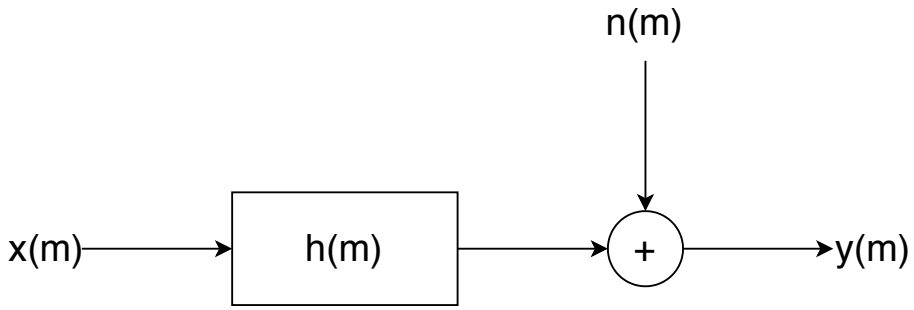


***Figure 2.2.*** *A model of environmental distortions.*

In mathematical notation, the model is formulated as

$$y(m) = x(m) * h(m) + n(m), \tag{2.1}$$

where $y(m)$ is the distorted signal, $x(m)$ is the clean signal, $h(m)$ is the convolutional noise or linear channel, $n(m)$ is the additive noise, $m$ is the discrete time index and $*$ denotes convolution. Discrete time is used in the model because it is assumed that the incoming signal $x(m)$ is the perfectly digitized version of the ideally recorded signal to make it possible to attribute also non-environmental distortions to the same noisy channel for simplicity. Considering only the environmental distortions in this model, convolutional noise $h(m)$ is a linear time-invariant filter that models the reverberation and spectral shaping effects of the environment. It can be estimated with a room impulse response (RIR), which can be measured with techniques described in Section 2.6.2. The additive noise $n(m)$ can be any background noise, but in further calculations it is often assumed to be a stationary perturbation and uncorrelated with $x(m)$. Therefore, in power spectral domain it holds that [36]

$$P_Y(\omega_k) = |H(\omega_k)|^2 P_X(\omega_k) + P_N(\omega_k), \tag{2.2}$$

where $P_Y(\omega_k)$, $|H(\omega_k)|^2$, $P_X(\omega_k)$ and $P_N(\omega_k)$ are the power spectra of the distorted signal, linear channel, clean signal, and additive noise, respectively, and $\omega_k$ is a particular frequency band. Since features used in audio analysis, such as MFCCs, are commonly derived from such spectra, noise can cause a data-mismatch error between the training and test sets in learning scenarios [1], which degrades the performance of pattern recognition systems significantly.

Besides environmental distortions, similarly a recording device can distort a signal during its capture. All microphones have their own non-ideal frequency responses, which affect a signal the same way as the linear channel described above. This means that the microphone attenuates certain frequencies, while ideally the frequency response would be flat and no attenuation would occur. In addition, the capture process may cause several other kinds of distortions such as clipping, aliasing, and data loss [55, Chapter 3].

The frequency response of the high quality microphone used in this work for room impulse response measurements is available at the webpage [13] of Earthworks Audio. Although the response is mostly flat, there is some minor deviation below 10 Hz and above 10 kHz.

Smartphone manufacturers do not usually publish the microphone frequency responses of their devices. A company, which develops measurement software for smartphones, measured frequency responses of three Apple devices [14]. The measured devices were iPhone 3GS, iPhone 4 and iPad. The responses are significantly worse than the response of the high quality microphone due to the lower quality of the microphones in the devices. The behaviour of the curves below 200 Hz and above 4 kHz is quite unpredictable. However, for the human voice frequencies the responses are almost flat, which is sufficient for the normal use cases of the smart device microphones.

## 2.3 Robust classification

In robust classification, the aim is to minimize the effect of noise on the performance of a classification model. In this work, the focus is on robustness to noise and distortions in the audio data. In other words, a model is robust, when it is capable to perform well even when data to be classified is noisy or distorted.

As machine learning techniques have recently been developing rapidly, robust classification has also gained attention due to its importance when working with noisy real-life data. A large number of studies have been made about improving noise robustness in audio analysis problems, especially in speech recognition [1, 2, 26, 36, 55] and sound event detection [31, 32, 35].

There are three main strategies [20] to improve noise robustness: usage of noise resistant features, signal enhancement, and model compensation for noise. Although the strategies are focused on noise robustness of speech recognition models, they may also be applied on other kinds of tasks.

### 2.3.1 Noise resistant features

As mentioned in Section 2.1, in feature extraction, feature vectors are extracted from raw audio signals to remove unnecessary information. The use of noise resistant features stands for selecting only such features, which preserve the important information

while being invariant to noise, reverberations and other distortions or for example speaker related differences in speech recognition. Noise resistant features are obtained by performing task-related and carefully chosen transformations for the original signals.

Although MFCCs are widely used in audio analysis as features, they are not robust to noise [42]. Several modifications to MFCCs have been proposed to account for noise robustness among with new types of features such as gammatone frequency cepstral coefficients [58].

There are also techniques for removing the effects of noise and distortion from noisy features after feature extraction. These feature enhancement [55, Chapter 9] techniques tend to rely on the availability of parallel clean and noisy features and they attempt to estimate the clean features from noisy features by using joint probability distributions.

In [28], RNNs were used to denoise utterances for a speech recognition problem. More specifically, the model was trained to predict clean MFCCs from noisy MFCCs by using parallel clean and noisy training data with varying noise levels. When tested with data corrupted with seen noise types, the denoising model outperformed a SPLICE algorithm [9] based system, which attempts to model joint distributions between clean and noisy data. However, with unseen noise types, the SPLICE algorithm based system performed better.

### 2.3.2 Signal enhancement

In signal enhancement, the goal is to make noisy signals clean from distortions before feature extraction and this way prevent data mismatch errors. Signals that are recorded only with a single microphone can be enhanced using filters [55, Chapter 4]. A simple approach is to use voice activity detection to locate frames consisting only of noise and to drop them. More advanced techniques involve adaptive spectral gain functions which are mostly effective in removing additive noise. Such functions operate on the spectral decomposition of a signal, and therefore it is necessary to also be able to reconstruct the enhanced time-domain signals afterwards without significant errors.

When dealing with multi-channel signals, it is possible to use a technique called *beamforming* [6], which can utilize also spatial information. Although it requires prior knowledge of the positions of the microphones in the microphone array used to capture the signals, it has the capability of tracking sound sources and it is also more powerful in reducing noise than single-channel enhancement techniques.

### 2.3.3 Model compensation for noise

The third approach to improve robustness concentrates on adjusting the classifier instead of enhancing the noisy test data. In speech recognition, one approach is to modify the parameters of the acoustic model [20], which maps utterances to phonemes or words, to

match the characteristics of the noisy environment. In speaker adaptation, the model is adjusted based on the characteristic features of individual speakers.

In [3], parameters of a hidden Markov model (HMM) trained for speech recognition with noisy speech were estimated from an HMM model trained with clean data and knowledge of the acoustical environment. Using the estimated parameters, comparable results with a matched condition were observed.

Another widely used technique consists of contaminating the training data with noise [20], which removes the mismatch caused by clean training data and noisy test data. Such noise contamination procedures are also referred to as data augmentation techniques.

*Data augmentation* [54, p. 139] means extending the existing data by performing label-preserving transformations on it. These transformations do not modify the semantic content of the data, but introduce previously unseen variations into the data. Simple examples of data augmentation are background noise addition for audio data, and rotation for image data. Besides using data augmentation to create noisy data from existing clean data, it can also be used to create more data when there is not enough available. Moreover, additional data decreases the chance of overfitting and hence improves performance. Different augmentation techniques for audio data are discussed in the next section.

## 2.4 Audio data augmentation techniques

A large number of audio data augmentation techniques have been presented in the literature. These techniques modify for example the signal-to-noise ratios (SNRs), reverberation times, and pitch of the sounds. Some data augmentation techniques such as pitch shifting and time stretching are implemented for Python in `librosa` [30]. Others may require external data such as background noise recordings or impulse responses, although using them only requires simple addition and convolution operations.

To visualize the transformations performed in the various data augmentation techniques, waveforms and mel spectrograms of an example audio sample processed with the techniques are prepared. In Figure 2.3, the waveform and the mel spectrogram of an utterance consisting of the phrase "good night" are shown. This figure is used as a comparison for the effects of data augmentation techniques presented in this section. In all the visualized techniques, the same sample is used as the input.

In addition to presenting the existing augmentation techniques, outcomes from using them in various audio analysis tasks are also reported. Because multiple techniques are often used together, it is possible to make comparisons of their effectiveness for different tasks.

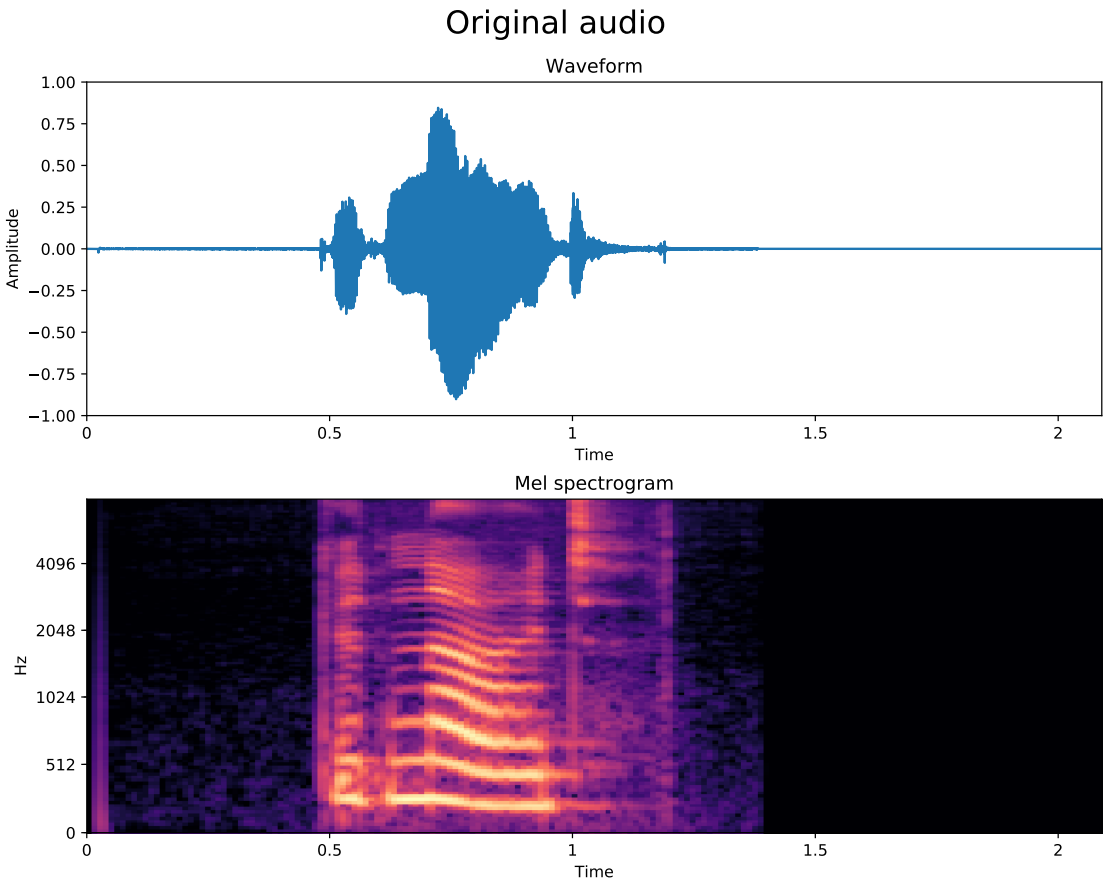


***Figure 2.3.*** *Original audio waveform and mel spectrogram.*

### 2.4.1 Additive noise

As its name suggests, *additive noise* is noise that is summed with the original signal. The type of noise can be for example Gaussian white noise, uniform random noise, or a background recording, such as an acoustic scene sample. The main difference between Gaussian white noise and an acoustic scene background is that the acoustic scene contains non-stationary events, which are expected to appear also in real noisy data. Implementing noise addition is simple since it requires only the summation of two signals, and the SNR of the output can be controlled by scaling the signals beforehand.

In Figure 2.4, Gaussian white noise is added to the original audio. The noise is equally distributed across all frequencies and it can be seen from the waveform as the stationary noise floor and in the spectrogram as the almost constant purple background.

In [47], it was found that even a small amount of additive Gaussian noise only increased the classification error in a singing voice detection task. Gaussian noise has not been lately used as much in augmentation of audio data as acoustic scenes, but it has been shown [4] to improve the generalization performance of other regression and classification problems.

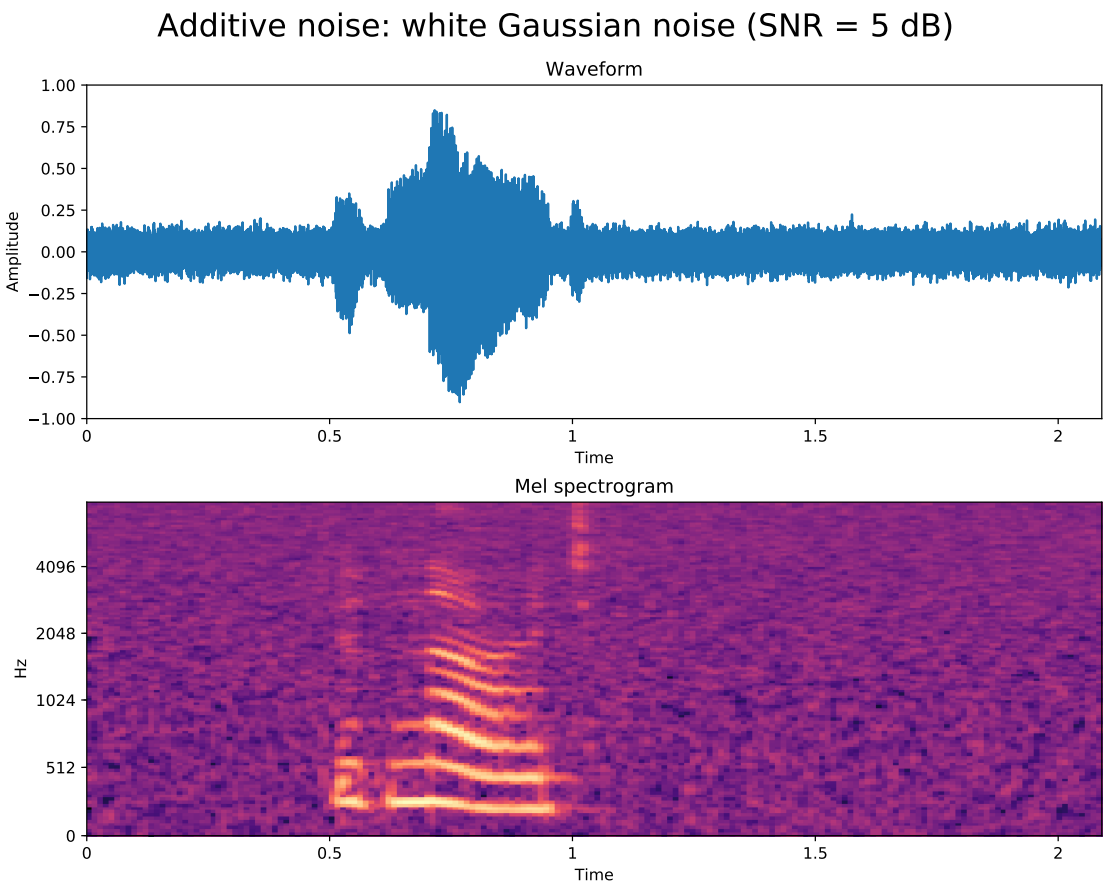


***Figure 2.4.*** *Additive white Gaussian noise.*

In Figure 2.5, an acoustic scene sample recorded in a restaurant is added to the original audio depicted in Figure 2.3 with an SNR of 5 dB. From the waveform it is visible that the added background makes the detection of the original signal quite difficult. On the other hand, in the spectrogram, the energy of the original signal is clearly standing out, and most of the noise is spread somewhat evenly across the frequency bins.

The use of additive acoustic scene recordings had a positive impact on the accuracy of an environmental sound classifier in [46]. Performance on some noise-like sound classes, such as an air conditioner, was reported to have been deteriorated however. The gain from using additive noise was highly dependent on the sound class overall, and a specific combination of augmentation techniques for each class was found to be the best solution.

Additive acoustic scenes did not improve significantly the performance of a musical instrument recognizer in [29]. Noise addition was used on top of other augmentation techniques, so the individual effect of additive noise was not reported. However, additive noise notably improved at least the recognition accuracy in case of vocalists and synthesizers.

Background noise consisting of different types of music, technical noises and non-technical noises from the MUSAN Noise dataset [49] was used in [27] to augment speech data from the LibriSpeech [37] dataset. When tested against clean test data, additive noise lowered the character error rate only marginally. Additive noise still outperformed the

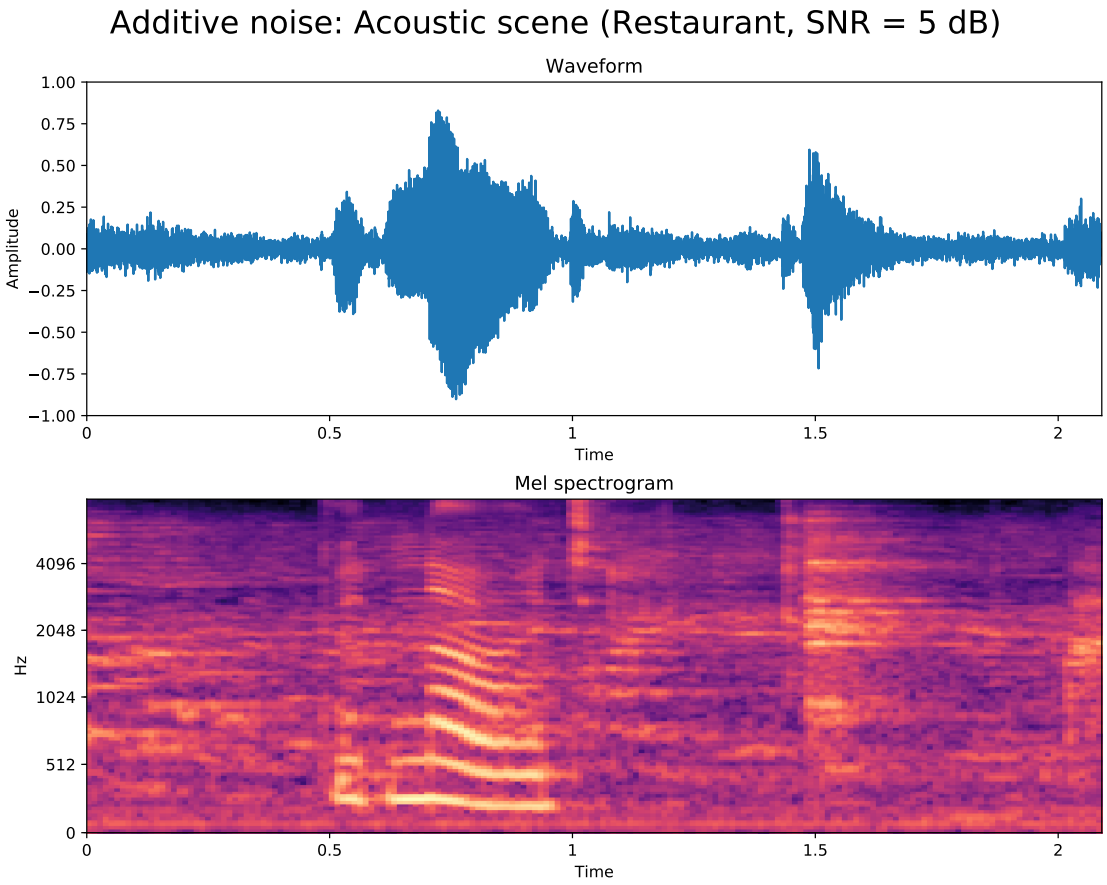


***Figure 2.5.*** *Noise addition using an acoustic scene recording.*

baseline when evaluating with noisy data, and especially when the test data was mixed with speech from other sources, i.e. in a multi-speaker environment.

### 2.4.2 Convolution with impulse responses

Convolution is an operation which can be used for filtering and cross-synthesis of signals. Cross-synthesis [44] emphasizes mutual frequencies in two signals and minimizes others, and in time domain it can affect the hanging time of specific frequency components, for instance. Convolving a signal with an impulse response of a linear and time-invariant (LTI) system is a type of cross-synthesis, where the characteristics of the system are imposed on the input signal. In practice, such a system can be for example a room where the characteristics define its reverberation time and other factors. Impulse responses are discussed more in detail in Section 2.6.

In mathematical terms [39, pp. 47–50], convolution for continuous-time signals (convolution integral) is defined as

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau, \tag{2.3}$$

and for discrete-time signals (convolution sum) as

$$y(n) = x(n) * h(n) = \sum_{i=-\infty}^{\infty} x(i)h(n-i), \tag{2.4}$$

where $y$ is the output signal, $x$ is the input signal, $h$ is the impulse response, $t$ is the continuous-time index, $n$ is the discrete-time index, and $*$ denotes convolution.

In frequency domain, convolution can be expressed as a simple multiplication, for a continuous case as

$$x(t) * h(t) = \mathcal{F}^{-1}\{X(\omega)H(\omega)\} = \mathcal{F}^{-1}\{\mathcal{F}\{x(t)\}\mathcal{F}\{h(t)\}\}, \tag{2.5}$$

and for a discrete case as

$$x(n) * h(n) = \mathcal{F}_d^{-1}\{X(k)H(k)\} = \mathcal{F}_d^{-1}\{\mathcal{F}_d\{x(n)\}\mathcal{F}_d\{h(n)\}\}, \tag{2.6}$$

where $\omega$ and $k$ denote continuous and discrete frequencies of the frequency domain, and $\mathcal{F}$ and $\mathcal{F}_d$ are continuous and discrete Fourier transform operators, respectively. If the signals are long, convolution in time domain quickly becomes computationally heavy. Therefore, it is often more practical to use the fast Fourier transform (FFT) to get to the frequency domain and do the operation there.

In Figure 2.6, convolution with a room impulse response is performed on the input signal of Figure 2.3. The room where the impulse response was measured is a highly reverberant bomb shelter. In time domain, the beginning of the signal is unchanged due to the silence, but the end of the signal has been extended due to increased reverberation in the signal. In frequency domain, the energy in the frequency bins has spread over the time axis, also because of the reverberation.

Room impulse responses were beneficial for a speech recognition task in reverberant environments in [43]. The word error rate (WER) was reduced from 59.7 % to 41.9 % for the IWSLT 2013 evaluation set by convolving the training data with impulse responses collected from various rooms. However, when testing against non-reverberant data, convolving the training data similarly increased the WER from 19.1 % to 26.2 %.

In [24], it was found that real room impulse responses yielded better results than simulated room impulse responses on a speech recognition task with several evaluation sets consisting of reverberated speech. When adding point-source noise to the augmentation routine, the performance gap between simulated and real impulse responses vanished. It was also noted that combining clean and augmented data in the training set was more useful than using only augmented data.

Using simulated room impulse responses created from very basic room information improved also the performance in speaker identification and mood detection tasks [10].

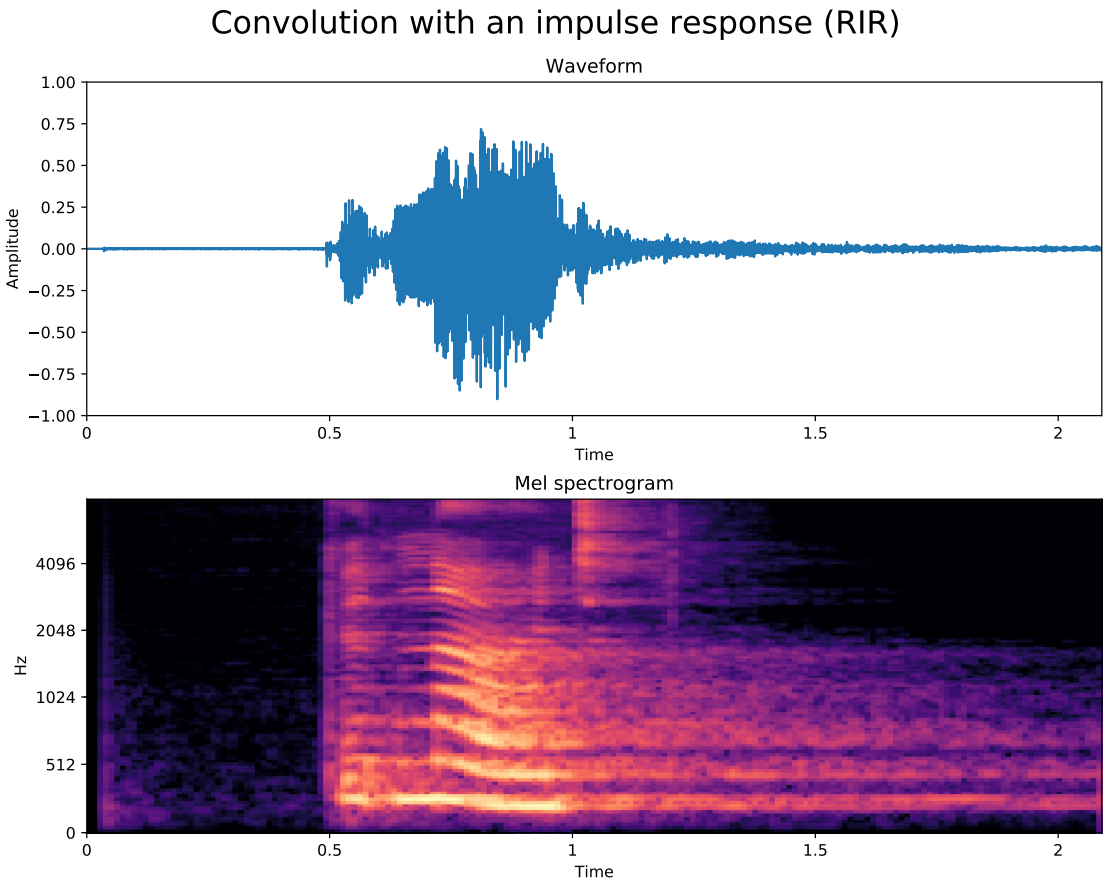


***Figure 2.6.*** *Convolution with a room impulse response.*

The evaluation data was collected in real reverberant environments and the system was capable of performing within 5 % – 10 % of a non-reverberant baseline.

An impulse response from the microphone of a Google Nexus One smartphone together with a room impulse response were used for convolutions in [7] for a musical instrument recognition task. For seven out of the twelve instruments in the task, the two-step convolution technique improved the performance of the recognizer over a nonaugmented baseline. For the majority of the instruments, other augmentation techniques improved the performance of the recognizer more than the convolutions. Since only one device and one room impulse response were used for the convolutions, robustness against new devices or rooms was not tested. Furthermore, the results from convolutions with only the smartphone microphone or the room impulse response was not reported.

### 2.4.3 Pitch shifting

In *pitch shifting*, all the frequency components in a sample are shifted upwards or downwards by a constant factor, making the audio sound higher or lower, while keeping the duration intact. This can be achieved in the frequency domain by scaling the linear-frequency spectrograms vertically, i.e. in the frequency dimension. Another approach

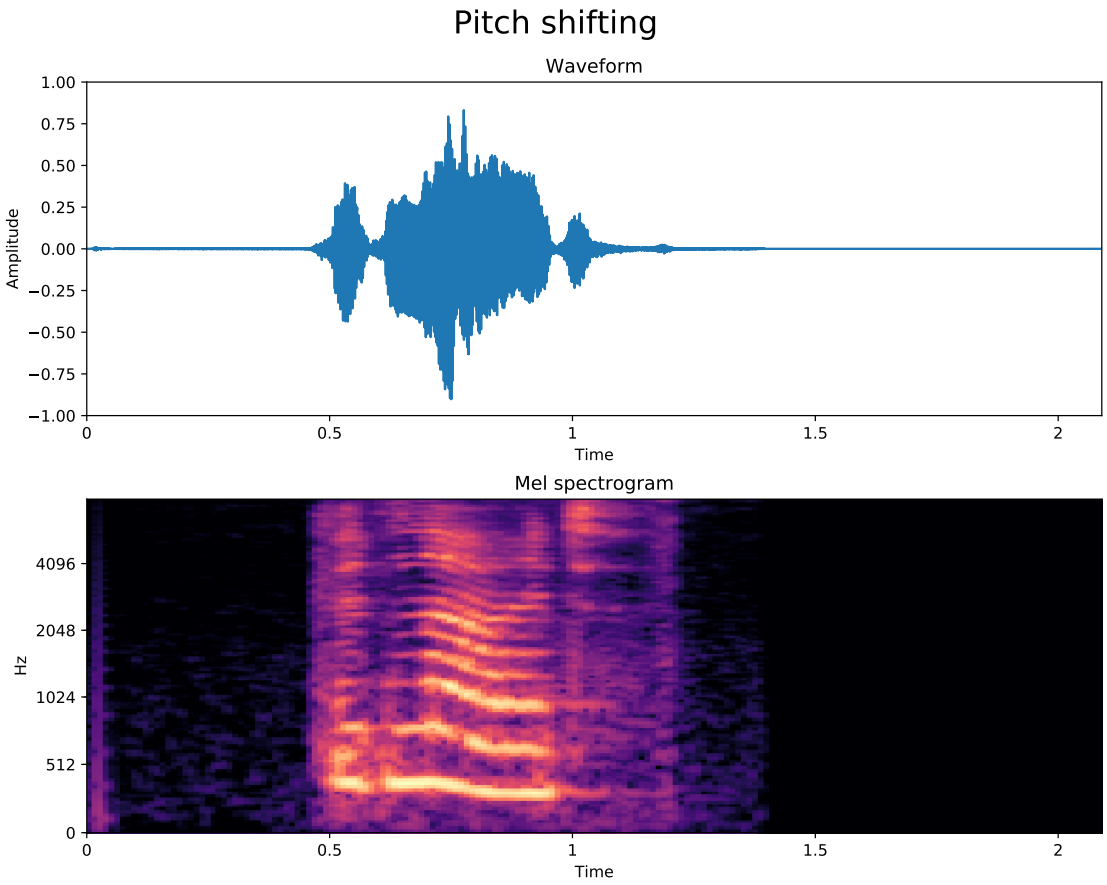


***Figure 2.7.*** *Pitch shifting by 6 semitones upwards.*

is to first stretch the sample in the time dimension and then resample, as was done in `librosa`. It has to be noted that pitch shifting upwards moves energy above the Nyquist frequency [56] of the sample and the energy is lost when reconstructing the waveform.

In Figure 2.7, pitch shifting upwards by six semitones has been performed on the example sample. The spectrogram reveals that the energy on the frequency bands has risen towards higher frequencies. The waveform has also changed shape due to the difference in wavelengths and loss of high frequencies.

Pitch shifting by $\pm 20\%$ or $\pm 30\%$ provided the most gain out of all the augmentation techniques compared in a singing voice detection task [47]. It reduced the classification error by 25 % on two separate evaluation sets consisting of single and multi-genre music snippets. In [46], pitch shifting was the most beneficial for sound event classification. It was also the only technique that did not have a negative impact on any of the classes.

### 2.4.4 Time stretching

In *time stretching*, the duration is scaled by a coefficient while retaining the original pitch of the sample. Time stretching can be done similarly as pitch shifting by scaling a linear-

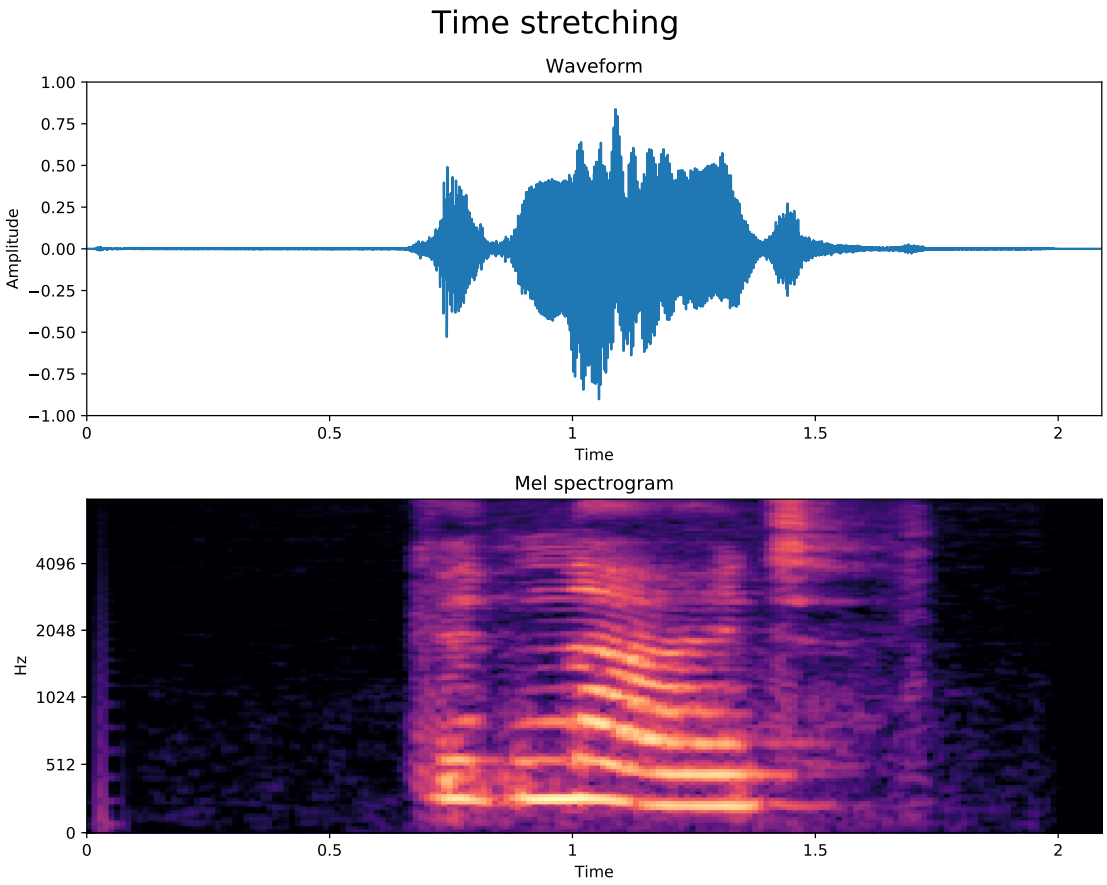


***Figure 2.8.*** *Time stretching by a coefficient of 0.7 (70 % speed of original).*

frequency spectrogram, but it is performed in the time dimension. Phase vocoding [17] is also used for pitch shifting and time stretching. It reduces the amount of artefacts in the resynthesized sounds by taking into account also phase information instead of just frequencies. For example in `librosa`, time stretching is performed with phase vocoding.

In Figure 2.8, the example sample has been stretched in time. The energy is spread out on the time axis in the spectrogram, but the energy is still in the same frequency bins. The waveform is also a stretched version of the original.

Time stretching has not been as successful as many other data augmentation techniques in the literature. In a music information retrieval task [29] it was found to be actually detrimental for classes such as synthesizer, violin, or female singer due to unnatural distortion of vibrato characteristics. Time stretching was on average capable of increasing the performance of a sound event classifier [46], although the gain was smallest of the tested techniques including pitch shifting, background noise, and dynamic range compression.

### 2.4.5 Vocal tract length perturbation

*Vocal tract length perturbation* (VTLP) is a data augmentation technique mostly used in speech recognition. Vocal tract length [25] determines spectral characteristics of speech. It is inversely proportional to the positions of spectral formant peaks in utterances for given sounds. Therefore, estimating and modifying these formant frequencies allows the normalisation and perturbation of vocal tract lengths among sets of speakers. A warp factor, $\alpha$ [22], is used to define the amount of perturbation and it maps center frequencies in mel scale filter banks to new frequencies. The mapping is performed with the function

$$f' = \begin{cases} f\alpha & f \leq F_{hi}\frac{\min(\alpha,1)}{\alpha} \\ S/2 - \frac{S/2-\min(\alpha,1)}{S/2-\frac{\min(\alpha,1)}{\alpha}} & \text{otherwise} \end{cases} \tag{2.7}$$

where $S$ is the sampling frequency and $F_{hi}$ is the upper boundary frequency limiting the chosen formants. The mel scale filter banks are then used as usual to create the mel spectrograms for feature extraction.

In [22], phoneme error rate was successfully decreased by using VTLP on the TIMIT dataset [18] in a speech recognition task. Improvements of at least over 0.5 %-points over non-augmented training baselines were achieved with all hyperparameter settings.

A speech recognition system for low resource languages [40] was evaluated with supervised and unsupervised learning settings with and without VTLP. The best results were achieved with a combination of a supervised nonaugmented Gaussian mixture model and a supervised VTLP-augmented multi-layer perceptron.

### 2.4.6 Dynamic range compression

In *dynamic range compression* (DRC), the dynamic range of an audio signal is reduced so that quiet sounds are amplified and loud sounds are attenuated. DRC was used in [29] with pitch shifting, time stretching, and background noise addition for instrument recognition. Compression was performed with speech and music settings defined in the Dolby E standard and it was implemented using the library `sox`. Increase in performance was observed only for the recognition of the following instruments: male singer, drum set, clean electric guitar, and distorted electric guitar. With other instruments, the performance was equal or lower than without DRC.

In [46], it was found that DRC was the most helpful technique in classification of gunshots, which typically consist of sudden peaks, out of all the sound events classified. However, DRC was most harmful for classifying noise-like air conditioner sounds.

### 2.4.7 Other techniques

Besides the aforementioned data augmentation methods, there are several techniques that are less frequently used. For example in [47], dropout, loudness, random frequency filters, and mixing were used in addition to the previously covered pitch shifting, time stretching, and Gaussian noise for a singing voice detection task.

*Dropout* was implemented like the neural network regularization technique with the same name, i.e. by setting inputs or spectrogram bin values to zero at a certain probability. In *loudness*, the spectrograms were simply scaled by a random factor to vary the energy levels in the frequency bins. *Random frequency filtering* consisted of creating and employing a large amount of filters with a Gaussian response and varying the values of $\mu$ and $\sigma$ randomly. Finally, in *mixing*, training examples were mixed with negative samples, i.e. samples without an active singing voice, and the resulting mix inherited the label of the training sample. The strength of the effect was controlled by a random scaling factor $f$ when summing the samples' spectrograms together. Out of these techniques, only random frequency filtering improved the performance of the detection system by a small amount. Loudness did not affect the performance, but dropout and mixing were found to be harmful.

*Blocks mixing* was also used in [38] to augment data for sound event detection. The mixing was done by combining different parts of a signal within the same context, i.e. scenes. For majority of the sound events, blocks mixing improved the F1 score of the system. Mixing was not beneficial in contexts such as beach and office, while in a car and a stadium it improved the performance considerably.

*Speed perturbation* was used in [23] with VTLP and time stretching (*tempo perturbation* in the paper) in training a speech recognition system. Speed perturbation was performed by resampling, which also affects the pitch unlike in time stretching, where the pitch remains unchanged. Speed perturbation was found to lower the WER more than the other tested techniques.

*Stochastic feature mapping* (SFM) was implemented in [8] to improve speech recognition of small languages with limited data. SFM is a voice conversion technique, which means that statistical characteristics of one speaker's speech are used to modify another speaker's utterance, making it possible to increase the amount of utterances from certain speakers. In most test cases, SFM yielded a lower WER than VTLP, although both of them increased the performance of the system by several %-points.

A *GSM coder* was used in [12] to emulate phone line channel effects on clean speech data with added background noise. The augmented data was used to train a whispering detector system, which reached an accuracy of 91.8 %. However, a comparison with a nonaugmented case was not performed.

*Multiple-width frequency-delta* (MWFD) data augmentation was presented in [21] and tested in an acoustic scene classification task. Delta features were extracted from spec-

trograms with varying widths to create additional data samples. MWFD with a convolutional neural network beat the compared baselines in nearly all acoustic scenes excluding only the café/restaurant and the grocery store scenes.

## 2.5 Datasets for audio data augmentation

To perform noise additions and impulse response convolutions, datasets of background recordings and impulse responses are needed. Collecting such data is a time-consuming process, and therefore using existing datasets is a valid option. When creating a system robust to realistic environmental distortions, a common choice is to use acoustic scene recordings as the added noise.

The availability of public impulse response datasets is somewhat lower than with acoustic scenes, but there are still some options to choose from. Their measurement is more complicated than collecting background noises, which may affect their availability.

### 2.5.1 Acoustic scene datasets

An acoustic scene is an environment that has a typical audio background which characterizes it and separates it from other locations. Examples of acoustic scenes are a restaurant, a library, or the inside of a bus. Mixing such recordings to the training data of an audio classifier is expected to make the system more robust to realistic environmental distortions. There are some acoustic scene datasets publicly available, although there is considerable variance in their quality and size. Although a large amount of background noise data is desirable for data augmentation purposes, the amount and selection of classes is also an important factor. Specifications of some of the largest available acoustic scene datasets are summarized in Table 2.1.

***Table 2.1.*** *Available acoustic scene datasets.*

| Dataset name | Classes | Examples | Size | Sr (Hz) |
|---|---|---|---|---|
| Dares G1 | 28 | 123 | 2 h 3 min | 44100 |
| DCASE 2013 Scenes | 10 | 100 | 50 min | 44100 |
| LITIS Rouen | 19 | 3026 | 25 h 13 min | 22050 |
| TUT Acoustic Scenes 2016 (DCASE2016) | 15 | 1170 | 9 h 45 min | 44100 |
| TUT Acoustic Scenes 2017 (DCASE2017) | 15 | 4680 | 13 h | 44100 |
| TUT Acoustic Scenes 2018 (DCASE2018) | 10 | 8640 | 24 h | 44100 |
| UEA Noise DB / Series 1 | 10 | 10 | 40 min | 22050 |
| UEA Noise DB / Series 2 | 12 | 35 | 2 h 55 min | 8000 |

As the table shows, DCASE[1] challenges have been a big contributor for audio scene datasets in the past few years. Besides them, only the LITIS Rouen dataset exceeds in length and number of examples. The selection of acoustic scene datasets for the data augmentation system in this work is further motivated in Section 3.2.

### 2.5.2 Impulse response datasets

Available impulse response datasets are listed in Table 2.2. Only the free datasets are presented here, but there are also additional databases that require a purchase and are often distributed with mixing software.

***Table 2.2.*** *Available room impulse response datasets.*

| Dataset name | Rooms | Measurement technique |
|---|---|---|
| ACE Corpus | 7 | Exponential Sine Sweep |
| AIR Database | 4 | Maximum Length Sequence |
| C4DM RIR Data Set | 3 | Exponential Sine Sweep |
| MARDY | 1 | Maximum Length Sequence |

As can be seen from the table, the number of available impulse response datasets is low. Furthermore, all of the datasets consist of only room impulse responses. The total number of impulse responses is not reported for any of the datasets, but in each dataset, there are multiple impulse responses from different locations measured with varying equipment from the rooms specified.

## 2.6 Impulse response measurement techniques

An impulse response $h(t)$ [39, pp. 71–76] is the output of an LTI system when the input to the system is an impulse, which is theoretically a signal with zero duration, infinite height (technically undefined) and an area of one. The impulse, or Dirac delta function [48, pp. 289–293], is therefore defined as

$$\delta(t) = \begin{cases} 0, & t \neq 0 \\ \text{undefined}, & t = 0 \end{cases} \tag{2.8}$$

which is constrained by

$$\int_{-\infty}^{\infty} \delta(t)\, dt = 1. \tag{2.9}$$

[1] `http://dcase.community/`

The Fourier transform of an impulse response $h(t)$ is the frequency response $H(\omega)$. The frequency response [15] determines how different frequency components are affected by the system. Because an impulse by definition contains all frequencies, the frequency response provides complete information of the system's tendency to amplify or attenuate any frequency, and the shift of phase for each frequency. Therefore, a frequency response is a more intuitive description of a linear system than an impulse response, although they both contain the same information.

Impulse responses are used to characterise the behaviour of LTI systems. In audio signals, they can for example contain the acoustic characteristics of rooms such as reverberation time, or information about the capabilities of loudspeakers or microphones to playback or capture signals correctly.

Two impulse responses measured in a bomb shelter and a small office are shown in Figure 2.9. Impulse responses consist of series of spikes that are caused by the direct sound from the source to the receiver and the subsequent reflections from surrounding surfaces.

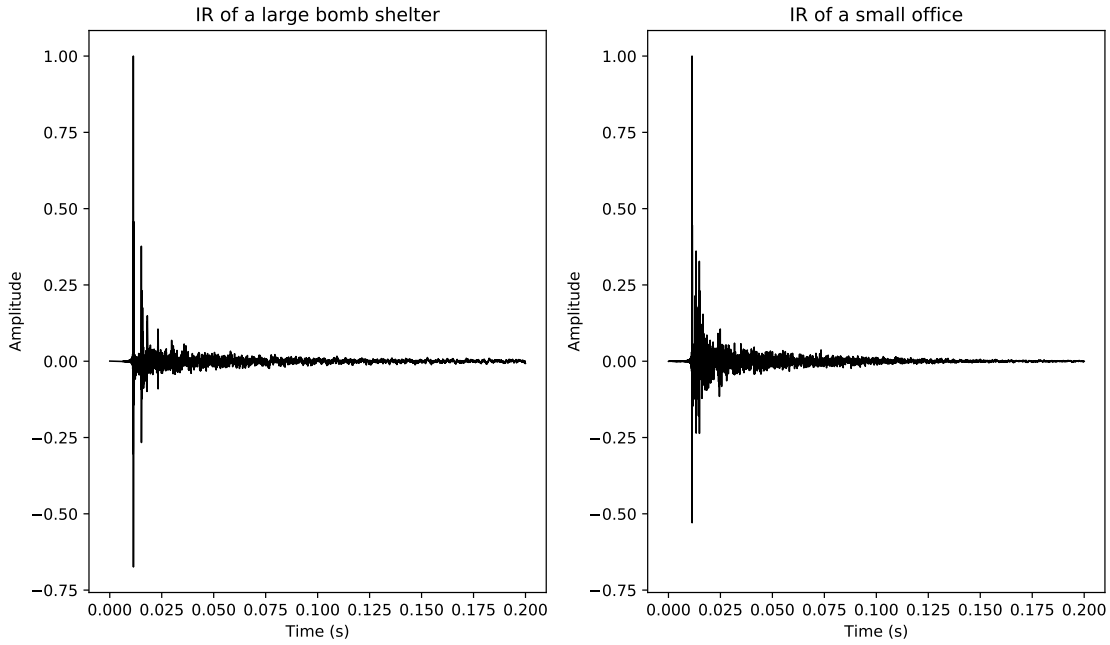


***Figure 2.9.*** *RIRs measured in a large bomb shelter and a small office.*

In the figures, there is a large spike in the beginning with another notable but much smaller one right after it as expected. The second spike is the result of the sound being reflected from the nearest surface, e.g. a wall. Rest of the impulse response is a combination of a large number of reflections from all directions. Perceptually, the loudness of the sound is increased by the early reflections, but the later reverberation reduces its intelligibility [39, p. 35]. Since the location in the left figure is a bomb shelter, the reflections last much longer than for example in the office in the right figure. This is visible from the amount of distortions in the tail of the bomb shelter IR. On the other hand, there are more obstacles in the office, which causes more spikes in the beginning of the office IR.

There are several techniques designed for measuring impulse responses of acoustic and audio systems. The most popular techniques for impulse response measurements are the exponential sine sweep (ESS) and the maximum length sequence (MLS).

### 2.6.1 Exponential sine sweep

The exponential sine sweep technique [16], also known as Farina method for its inventor, was designed for measuring impulse responses of acoustic systems that are not exactly LTI systems but close. Unlike the commonly used MLS technique, ESS tolerates minor nonlinearities and time-variances well and is overall more robust for distortions during the measurement.

First, a sine sweep, i.e. the excitation signal, is constructed. The sweep is defined as

$$x(t) = \sin\left[\frac{\omega_1 \cdot T}{\ln\left(\frac{\omega_2}{\omega_1}\right)} \cdot \left(e^{\frac{t}{T}\cdot\ln\left(\frac{\omega_2}{\omega_1}\right)} - 1\right)\right], \tag{2.10}$$

where $T$ is the duration of the sweep in seconds, $\omega_1$ is the starting lower frequency, and $\omega_2$ is the ending higher frequency. Then, an inverse filter $f(t)$ is generated by time-reversing the excitation signal and applying an envelope on it, which starts from 0 dB and ends at $-6 \cdot \log_2\left(\frac{\omega_2}{\omega_1}\right)$. Because now

$$x(t) * f(t) = \delta(t), \tag{2.11}$$

where $\delta(t)$ is the Dirac delta function, and

$$x(t) * h(t) = y(t), \tag{2.12}$$

where $h(t)$ is the impulse response of the system to be measured, we get

$$h(t) = y(t) * f(t). \tag{2.13}$$

Therefore, playing and recording the sine sweep in a room and simply convolving the recorded signal with the inverse filter yields the impulse response.

ESS is sensitive for noise, which needs to be taken into consideration when choosing a room to measure. However, ESS is capable of producing valid impulse responses even if there are unwanted harmonics in the excitation signal. The harmonics create smaller copies of the real impulse response that appear in the calculated $h(t)$ one after another, which makes it possible to simply cut them off afterwards. Furthermore, the SNR of the ESS technique is by far the highest out of the impulse response measurement techniques

presented in the literature [50]. In this context, SNR is the ratio between the power of the recorded signal and the power of the noise in the tail of the calculated impulse response.

### 2.6.2 Maximum length sequence

Maximum length sequence [19] is a pseudorandom binary sequence, whose autocorrelation function approaches a unit impulse when the length of the sequence increases. Due to this property, it can be used for measuring impulse responses of LTI systems. The cross-correlation of the recorded sequence $y(n)$ and the sequence $s(n)$ itself is

$$\phi_{sy} = h(n) * \phi_{ss} = h(n) * \delta(n) = h(n), \tag{2.14}$$

where $\phi_{sy}$ denotes cross-correlation between $s(n)$ and $y(n)$, $\phi_{ss}$ is the autocorrelation of $s(n)$, and $\delta(n)$ is the unit impulse, i.e. the discrete counterpart of the Dirac delta function.

Although the MLS technique loses to ESS in SNRs and for its strict linearity requirements, it handles background noise better during measurements [50]. Therefore, if there are people in the room that needs to be measured, MLS would be the better option.

# 3 METHODS

In this chapter, the tool for audio data augmentation (TADA) created for this work is introduced and the steps for implementing it are defined. First, the selection of the augmentation techniques for the tool is motivated. Next, the actual implementation of the augmentation techniques is described and further specifications of the tool are presented. Finally, the collection process of necessary augmentation data is explained.

## 3.1 Tool for Audio Data Augmentation

TADA is a tool for augmenting audio data for classification purposes. It was designed specifically for simulating the effect that a sound undergoes when it is recorded with a mobile device in varying locations. The inspiration for this was to robustify a phoneme error recognizer operating with mobile device recordings, i.e. to widen the range of devices and locations when training the underlying classifier.

### 3.1.1 Motivation

Factors that affect the sound when it travels from the sound source to the recording device are the room itself, modeled by a room impulse response, and background noise. Furthermore, when the sound is captured with the device's microphone, it is affected by the microphone's and the amplifier's responses, which are not ideal. If the nonlinear internal processes of the microphone and the recording setup are not taken into account, the device can also be modeled by a simple impulse response. This leads into three distinct augmentation steps, which are convolution with the RIR, summation with additive noise, and finally convolution with the mobile device impulse response (Figure 3.1). The implementation of the augmentation steps is explained in more detail in Section 3.1.2.

To create TADA, a sufficient number of RIRs, additive noise samples, and device IRs are needed. Due to the absence of publicly available mobile device IRs and the desire to obtain IRs from some newer phone models, we decided to collect the IRs ourselves. To get experience of the impulse response collection process, the IR collection method was first tested with rooms because their IR measurements are simpler due to the lack of mobile device hardware and application related problems. Although there are some RIR datasets available, collecting them ourselves simplifies the evaluation process and

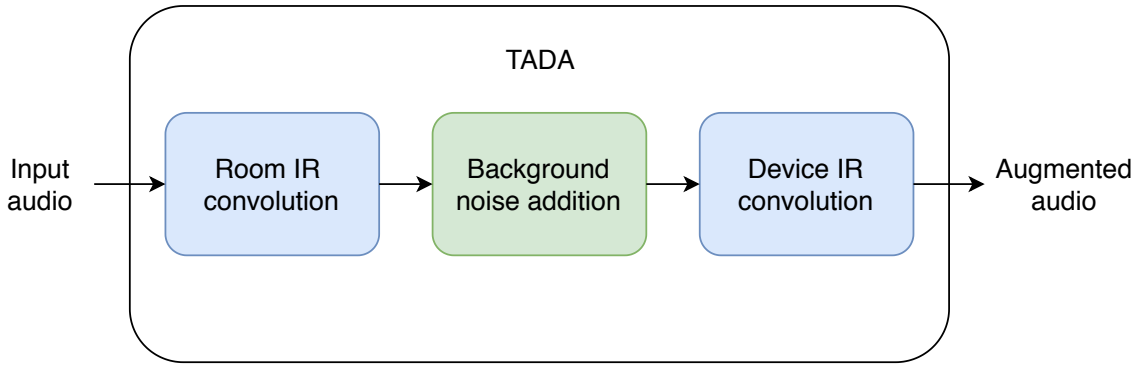


***Figure 3.1.*** *Flow diagram of the combined augmentation process.*

makes it easier to append the dataset with new rooms. Publicly available noise datasets on the other hand offer enough variation, and the process of selecting the datasets is described in Section 3.2.

### 3.1.2 Implemented augmentation techniques

Augmentation techniques implemented in TADA are addition of background noise with a variable SNR and convolution with room and device impulse responses. Each of the three steps can be stacked on top of each other and the processed sound will have the same length as the original. The augmentation process studied in detail in this work combines convolution with a room impulse response, addition of noise, and convolution with a device impulse response in this order to mimic the process of recording a clean sound with a mobile device.

In the noise addition, a noise sample is selected randomly from the chosen dataset(s) and a randomly chosen segment with the same duration as the input audio is cut from it. The segment is then scaled according to the desired SNR and summed with the input audio.

In the convolution method, an impulse response either from the room or device impulse response dataset is selected randomly. The convolution is then efficiently performed by multiplying the input audio and impulse response signals in the frequency domain using FFT.

### 3.1.3 Specifications

TADA was designed mainly for a cross-validation setup with five folds and training, validation, and test sets. This enabled artificial creation of noisy test data in order to evaluate the proposed method in addition to increasing the amount of training data. Because of this, the interface includes individual parameters, such as SNRs, for different subsets.

Because the split is only related to the evaluation of the system, it is explained in more detail in Section 4.3.1. Still, TADA can also be used to just augment training data, as is usually the case.

TADA was implemented with Python 3.6, and besides the standard library, the following packages were used: `glob2`, `numpy`, `pandas`, `scikit-learn`, `scipy` and `soundfile`. It is implemented as a class that offers methods for processing audio samples with the selected three augmentation techniques.

Initializing TADA with for example only the DCASE2017 background dataset, and selecting room and device impulse responses to be split by the recording position and the manufacturer of the device, respectively, the call looks like the following:

```
from augmenter import Robustifier
aug_file_folder = '~/Documents/data_augmentation'
robustifier_params = {'file_path': aug_file_folder,
                      'snrs': [-18, -12, -6, 0, 6],
                      'val_snrs': [-18, -12, -6, 0, 6],
                      'test_snrs': [0, 6, 12, 24, 48],
                      'datasets': 'dcase17',
                      'default_process': ['room', 'noise', 'phone'],
                      'room_split_by': 'position',
                      'phone_split_by': 'split_dimension',
                      'with_validation': True,
                      'random_seed': 42,
                      'single_set': None}

robustifier = Robustifier(**robustifier_params)
```

Here, `file_path` refers to the directory where the background recordings and impulse response files are located. The parameters `snrs`, `val_snrs`, and `test_snrs` are the target SNRs of the augmented training, validation, and test sets, respectively. The parameter `datasets` is used to specify the background noise datasets, and it is also possible to pass a list of datasets instead of just a single dataset. The parameter `default_process` determines the augmentation processes and their order, if the method `process()` is called. The parameters `room_split_by` and `phone_split_by` are used to select the method to split the room and device impulse responses for a cross-validation setup. The parameter `with_validation` controls the creation of a validation set for evaluation and `random_seed` the seed used to initialize the random number generators needed in selecting backgrounds and impulse responses randomly. To use TADA to augment data only in a single subset such as train, the `single_set` parameter is given the name of the desired subset.

Four methods were implemented for TADA: `convolve_room()`, `mix()`, `convolve_phone()` and `process()`. They have the following signature:

```
# convolution with a room impulse response
```

```
audio = robustifier.convolve_room(audio, **kwargs)

# noise addition
audio = robustifier.mix(audio, **kwargs)

# convolution with a phone impulse response
audio = robustifier.convolve_phone(audio, **kwargs)

# all processes combined
processes = ['room', 'noise', 'phone']
audio = robustifier.process(audio, processes=processes, **kwargs)
```

The keyword arguments are used to differentiate between subsets and split partitions when evaluating the system. Defining the parameter `subset` in any of the methods as the value of the `single_set` parameter passed for the constructor enables augmentation with all the available data.

## 3.2 Additive noise dataset collection

Acoustic scenes were selected as the type of additive noise for the system because the goal was to transform the clean data into being recorded in different locations. The available acoustic scene datasets (Table 2.1) were studied based on their number of classes, number of samples, and the sampling frequency they were collected with.

The datasets chosen for the implementation were the IEEE AASP Challenge Scene Classification dataset (DCASE2013) [52] and the TUT Acoustic scenes 2017, Development dataset (DCASE2017) [34]. They both offer a sufficient variety of scenes, which are listed in Table 3.1. Some of the scenes are overlapping, but overall the datasets complement each other well.

They were both also recorded with a sampling rate of 44100 Hz, which was the sampling rate of the audio data used in the evaluation system specified in the next chapter. In addition, DCASE2013 was recorded with different equipment than DCASE2017, which adds variation to the data. DCASE2016 dataset is a subset of DCASE2017, so it was left out. The datasets with a sampling rate other than 44100 Hz were not used in order to keep the quality of the augmented data as good as possible and not having to use upsampling in the process.

Since the chosen datasets were recorded with high quality binaural microphones, an additional background noise dataset recorded with mobile devices was collected by a third-party. The recordings are five-second long clips from various locations where people normally use their devices. The acoustic scenes were not controlled, so the distribution of the locations in the dataset is unknown. Still, this dataset is the most authentic choice for

***Table 3.1.*** *Acoustic scenes in the selected datasets.*

| DCASE2013 | DCASE2017 |
|---|---|
| Bus | Bus |
| Busy street | Café/Restaurant |
| Office | Car |
| Open-air market | City center |
| Park | Forest path |
| Quiet street | Grocery store |
| Restaurant | Home |
| Shop/Supermarket | Lakeside beach |
| Subway station | Library |
| Subway-train | Metro station |
| Urban park | Office |
| | Residential area |
| | Train |
| | Tram |

augmenting clean audio to transform it into mobile device recorded data, and therefore it was used exclusively in the evaluation stage in Chapter 4.

Addition of extra background noise datasets to TADA is possible, although cross-validation splits for evaluation are only designed for the chosen datasets. At minimum, a function for loading the file lists of the dataset into TADA is needed, but otherwise the system is capable of handling any dataset.

## 3.3 Impulse response measurements

Due to the lack of suitable impulse response datasets, we measured the room impulse responses and mobile device impulse responses ourselves. This allowed choosing the locations and the devices, and designing the IR datasets so that their usage in cross-validation was reasonable.

Impulse responses were measured with the exponential sine sweep method (Farina method) explained in Section 2.6.2. The measurement equipment and the parameters of the sine sweep for both room and mobile device measurements are in Table 3.2.

Audacity[1] was used as the recording software in all impulse response measurements. In device impulse response measurements, mobile applications were used to pass the data from the device microphone to the computer via the audio interface. The Extra Mic[2] app

[1] `https://www.audacityteam.org/`

[2] `https://play.google.com/store/apps/details?id=extra.chan.audio.extramic&hl=en_US`

***Table 3.2.*** *Impulse response measurement details.*

| | Room measurements | Mobile device measurements |
|---|---|---|
| **ESS parameters** | | |
| Sweep length (s) | 10 | 10 |
| Sweep range (Hz) | 80-20000 | 80-20000 |
| **Equipment** | | |
| Microphone | Earthworks Audio M30 | Mobile device |
| Loudspeaker | Genelec G Two | Genelec 1029A |
| Audio interface | Focusrite Scarlett 18i20 1st gen | Focusrite Scarlett 2i2 2nd gen |
| **Quantity** | | |
| Class count | 5 | 11 |
| Examples | 78 | 160 |
| **Classes** | | |
| | Small - Office (TC316) | Huawei Mate 10 lite |
| | Medium - TEK Lounge (TB110) | iPhone SE |
| | Medium - Meeting room (TE307) | iPhone 6S+ |
| | Large - Festia Great Hall (FA044) | iPhone 8 |
| | Bommari (Bomb shelter) | LG G4 |
| | | Motorola Moto C |
| | | Motorola Moto G (3rd gen) |
| | | Samsung Galaxy J5 |
| | | iPad Pro 12.9" |
| | | Headset 1 (iPhone 8) |
| | | Headset 2 (Huawei Mate 10 lite) |

was used with all Android devices, and the Megaphone[3] app with all Apple devices to enable the microphones. The bottom microphone of the devices was used for recording whenever it was possible to select the recording microphone. The quality of the recorded sweeps was confirmed with Audacity and the initial impulse response calculations during recording sessions was done in MATLAB. After all the measurements were completed, the impulse responses and metadata were postprocessed with Python.

### 3.3.1 Room impulse responses

Room impulse responses were measured in five locations in TUT: small office, medium living room/lounge, medium meeting room, large lecture hall, and a very large bomb shelter. The locations were selected so that there would be enough variation in their

[3] `https://apps.apple.com/us/app/megaphone-voice-amplifier/id304955183`

acoustical characteristics. The reverberation times varied from hundreds of milliseconds in small and medium rooms to several seconds in the bomb shelter.

Impulse responses were measured from five positions per room. First four positions were from each corner of the room and the fifth position in the center of the room. An exception was made with the bomb shelter because there were many different kinds of areas where to measure. Therefore, the measurements were made in a hallway and a hall from randomly picked spots instead of corners, which were difficult to define in such a space.

The placement of the microphone and the loudspeaker in rooms in RIR measurements is shown in Figure 3.2. With the corner measurements, the microphone was placed into the corner facing the room's center, and the loudspeaker 100 cm towards the center facing the microphone. The same distance between the loudspeaker and the microphone was retained also in the fifth position. The microphone and the loudspeaker were set to a height of approximately 1.5 m.

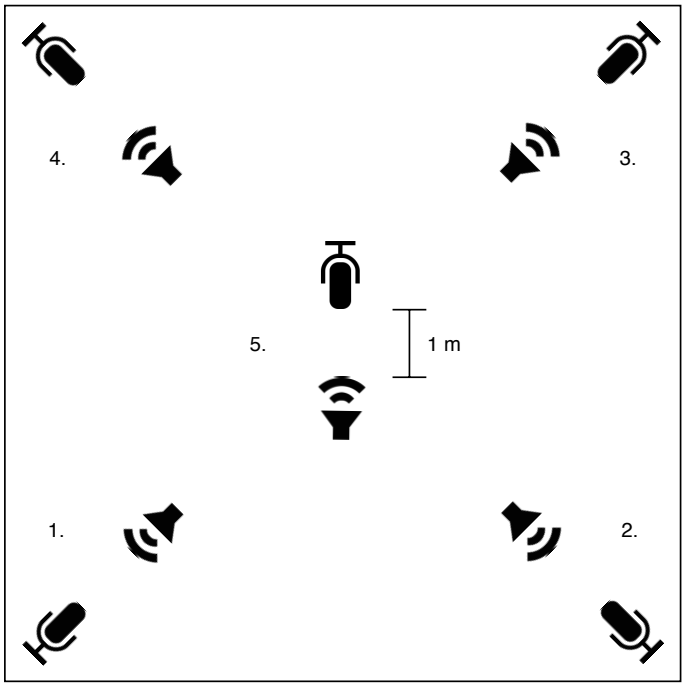


***Figure 3.2.*** *Placement of the microphone and the loudspeaker in RIR measurements.*

In each position, three measurements were made by varying the angle at which the loudspeaker was facing the microphone relative to the straight orientation. The angles of the three measurements were -15°, 0° and 15° as illustrated in Figure 3.3.

To make sure that the recorded sine sweeps capture the reverberations sufficiently, a sweep length of 10 seconds was used. The duration was based on the observation that only in the bomb shelter the audible reverberations lasted several seconds. A sweep range from 80 Hz to 20000 Hz was selected to cover most of the human hearable frequencies. The lower end was raised to 80 Hz from the usual 20 Hz due to limitations in the equipment to playback the lower frequencies.

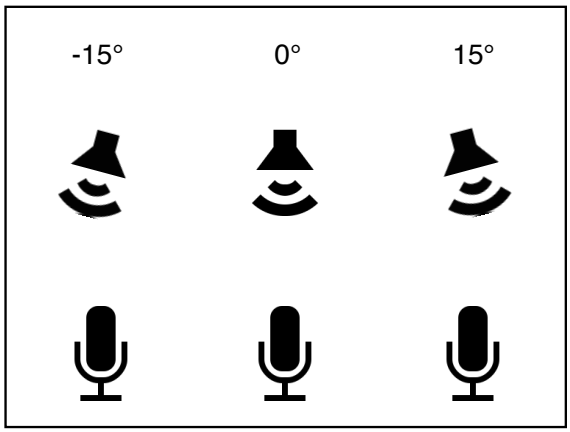


***Figure 3.3.*** *Directions of the loudspeaker in RIR measurements.*

### 3.3.2 Mobile device impulse responses

Mobile device impulse responses were measured with the same method as rooms but in an anechoic chamber so that the characteristics of the room would not affect the measurement. Microphone impulse responses from a total of nine mobile devices and two headsets were measured. The IRs were measured mainly from eight positions, but with some devices additional positions were used. Detailed list of the devices can be seen in Table 3.2.

The loudspeaker was put into the corner of the chamber facing the center in a similar manner as in Figure 3.2 but reversed. A person holding the mobile device was instructed to stand closely behind the loudspeaker, as if the speaker was his/her mouth. A person instead of a stand was used in the measurement also to account for the reflections from the person's body that would occur in real life scenarios. The person was asked to hold the device at a 30 cm distance from the loudspeaker facing it, on two different levels: at the level of the person's chest and at the level of the person's face. On both levels, the device was held in three different orientations (Figure 3.4): horizontal to the ground with screen facing up, 45° towards the loudspeaker, and vertical to the ground. In the figure, the arrow points to the direction of the screen. In addition, the person was instructed to sit at a table next to the loudspeaker holding the device first in hand, and then with the device resting on the table screen facing up.

Because the person causes reflections of sound when standing behind the loudspeaker, five people assisted in creating some variation in the amount and type of the reflections. The table was used in the measurements because it creates additional reflections and using a mobile device while sitting at a table was considered to be somewhat common.

During the measurements, it was found that the sound pressure level and the distance to the mobile device are crucial factors in the success of the measurements. The recorded sweeps often contained notable energy in the harmonic frequencies in addition to the actual frequency of the sine if either the sound pressure level was too high or the device was

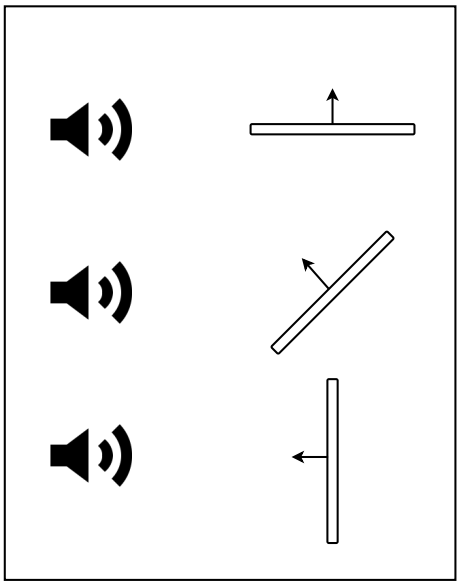

***Figure 3.4.*** *Directions of the mobile device in device IR measurements.*

too close to the loudspeaker. Therefore, it is essential to select the playback equipment carefully to reduce the probability of such issues.

Some of the mobile devices also had problems with aliasing even though the selected sampling rate should have prevented it. The devices may therefore have a fixed lower sampling rate and perform resampling afterwards. Automatic gain control can also affect the reliability of the impulse responses because it is assumed that there is no digital signal processing performed with the signals. Most of the time, Apple devices worked well in measurements. There were often problems with Android devices because of the large variation in quality of the microphones and software differences.

Mobile phones have nowadays often more than just one microphone built into them [5]. Because the direction of sound affects the impulse response, the outcome depends on the microphone used in the recording part of the measurement. Therefore, it is important to be aware of the number and location of the microphones beforehand. In addition, the recording application has to allow the selection of the microphone.

# 4 EVALUATION

To evaluate TADA, a classifier and a suitable audio dataset were needed. Since the focus was purely on the augmentation system, the classifier did not need to be optimized. Therefore, a classifier architecture and the data used in a pronunciation classification system [11] were used. Since in the paper fixed architectures were not used but optimized architectures for each evaluation case individually instead, one of the possible set of hyperparameters was selected for this work.

## 4.1 Data

The dataset consists of recordings of 120 mostly Finnish subjects pronouncing 80 different English words two or three times. The words were selected by English teachers so that they would cover most of the errors Finnish speakers tend to make when speaking British English. In this work, only a subset of five words was used to evaluate the effect of the augmentation system, and the words were *hit*, *job*, *join*, *pull* and *worse*. The words were picked based on the performance in [11] and the balance between correct and erroneous pronunciations.

The data was collected in a 4.53 m x 3.96 m x 2.59 m noise-insulated room with a reverberation time of 0.26 s. A Røde NT55 condenser microphone and a Focusrite Scarlett 2i2 audio interface were used to record the audio with a 40 cm distance from the speaker and a 44.1 kHz sampling rate.

The labels for the data contain the information of the recorded utterances being either pronounced correctly or with a specific type of error, *primary error*, defined for the target phoneme of the word in question. In addition, somewhat rare secondary errors were also labeled if present. Only the primary errors were used in the evaluation due to the small count of secondary errors, which resulted in a binary classification problem.

Primary errors for the phonemes in the selected words and their zero rule accuracies are presented in Table 4.1. Four different phonemes were targeted in the chosen words with their unique primary errors. Zero rule is the accuracy that would result from predicting the most common class in the dataset for all the samples. The average zero rule is used as one of the baseline scores when evaluating the classifier.

***Table 4.1.*** *Selected words, their primary errors, and zero rule accuracies.*

| Word | Phoneme | Primary error | Zero rule, % |
|---|---|---|---|
| hit | [ɪ] | [i] | 62.68 |
| job | [dʒ] | missing voicing | 77.52 |
| join | [dʒ] | missing voicing | 70.15 |
| pull | [p] | missing aspiration | 84.12 |
| worse | [w] | dentalisation | 68.56 |

## 4.2 Classifier

The architecture of the classifier used for evaluating TADA is presented in Figure 4.1. The classifier is an RNN consisting of three bidirectional long short-term memory (LSTM) layers with 100 nodes in each and a fully-connected output layer (Dense) with a single node. Bidirectional LSTMs were chosen because they allow information from both past and future frames of a sequence to be used. Since erroneous pronunciation in a specific part of a word might affect also rest of the word, this extra information can be beneficial.

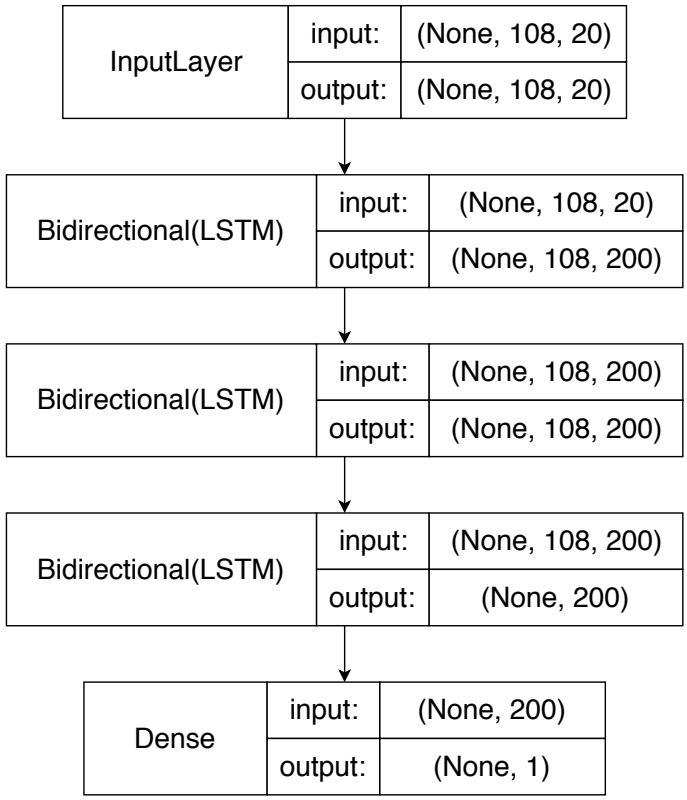


***Figure 4.1.*** *Classifier architecture.*

MFCCs with 128 mel bands and 20 coefficients were used as features for the audio data. After feature extraction, all of the data was standardized and padded with zeros to the maximum length of all the samples.

A five-run Monte Carlo cross-validation setup with 0.6/0.2/0.2 split proportions to training, validation, and test subsets was originally implemented to evaluate the classifier. To be able to compare results reliably, the random seed used for the splits was fixed.

Before the split, samples with disagreeing annotations were discarded. With two annotators, the ground truth for a disagreed sample would have been ambiguous and therefore measuring the performance would have become more complicated.

## 4.3 Experiments

To evaluate TADA, four experiments were designed: first an experiment for testing the effect of the room impulse responses, then another one for device impulse responses, one for additive noise, and finally an experiment for all of the three augmentation steps combined. Since all of the speech data used to train and test the classifier was collected in laboratory conditions, TADA was used to introduce noise also in the test data. Therefore, partitioning of the augmentation data was necessary to ensure that the data would not leak between the subsets of the word data.

### 4.3.1 Partitioning the augmentation data

The initial goal for the number of partitions in each impulse response and background noise dataset was set to five because the CV setup of the classifier was implemented with 5 Monte Carlo splits. DCASE2013 and DCASE2017 datasets are distributed with ready made splits into train and test subsets, but in DCASE2013 there was only one split and in DCASE2017 four splits available. DCASE2017 provides also the mapping between the samples and their original recordings. The original recordings are long continuous recordings from a single scene, which are typically cut into multiple shorter segments to increase the number of examples. The mapping allows the data to be split again so that the samples from the same recording do not end up in different subsets. Additionally, it is possible to partition the data based on the individual scenes of the original recordings. However, there was not enough information about the origins of the samples in the DCASE2013 to make new splits for it.

The background samples of the extra dataset recorded with mobile devices were split based on the unique user id's of the people recording the samples with their mobile devices as shown in Figure 4.2. The method is similar to the one used with DCASE2017, this time only using the user id's as the constraining groups instead of original recordings.

Several different split methods were designed for the impulse response datasets. The room impulse responses were split based on rooms and measurement points inside the rooms as illustrated in Figure 4.3. With five rooms and five points the number of partitions matches exactly the desired amount.

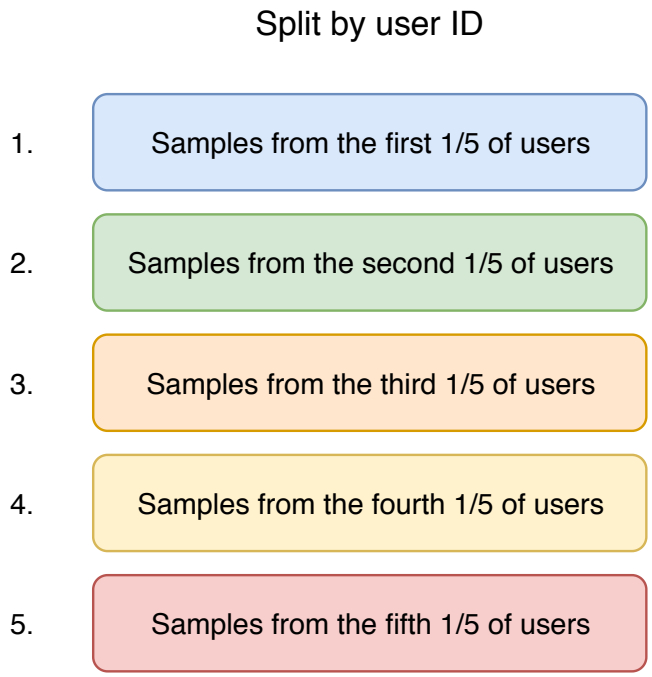


***Figure 4.2.*** *Partitioning of the background noise samples.*

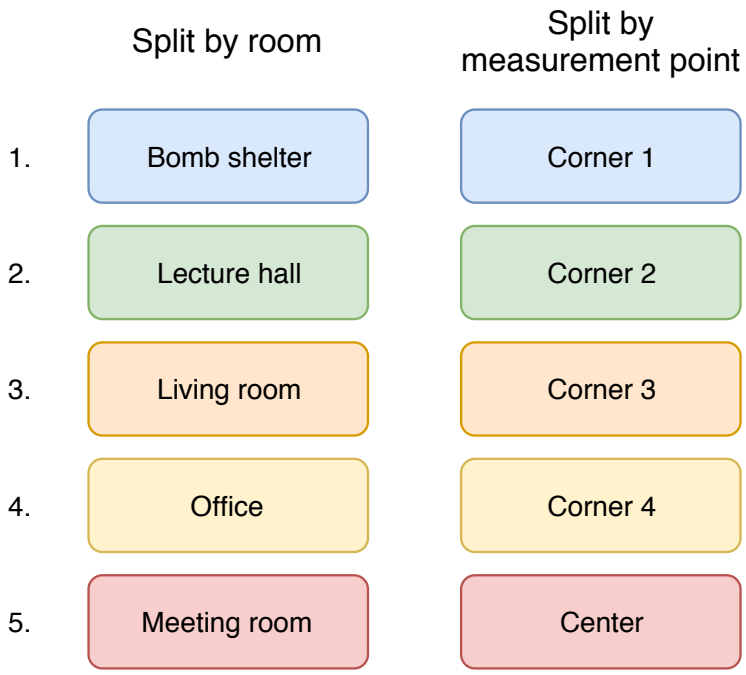


***Figure 4.3.*** *Partitioning of the room impulse responses.*

The device impulse responses offered more possibilities for ways of splitting. The implemented splits shown in Figure 4.4 were based on the manufacturer of the device, the model of the device, and the user holding the device in specific user measurements.

The manufacturer split is referred to as the *split dimension* in the code in Section 3.1.3 because headsets were put into their own separate categories despite them sharing a common manufacturer with some devices. This was done to enable testing the effects

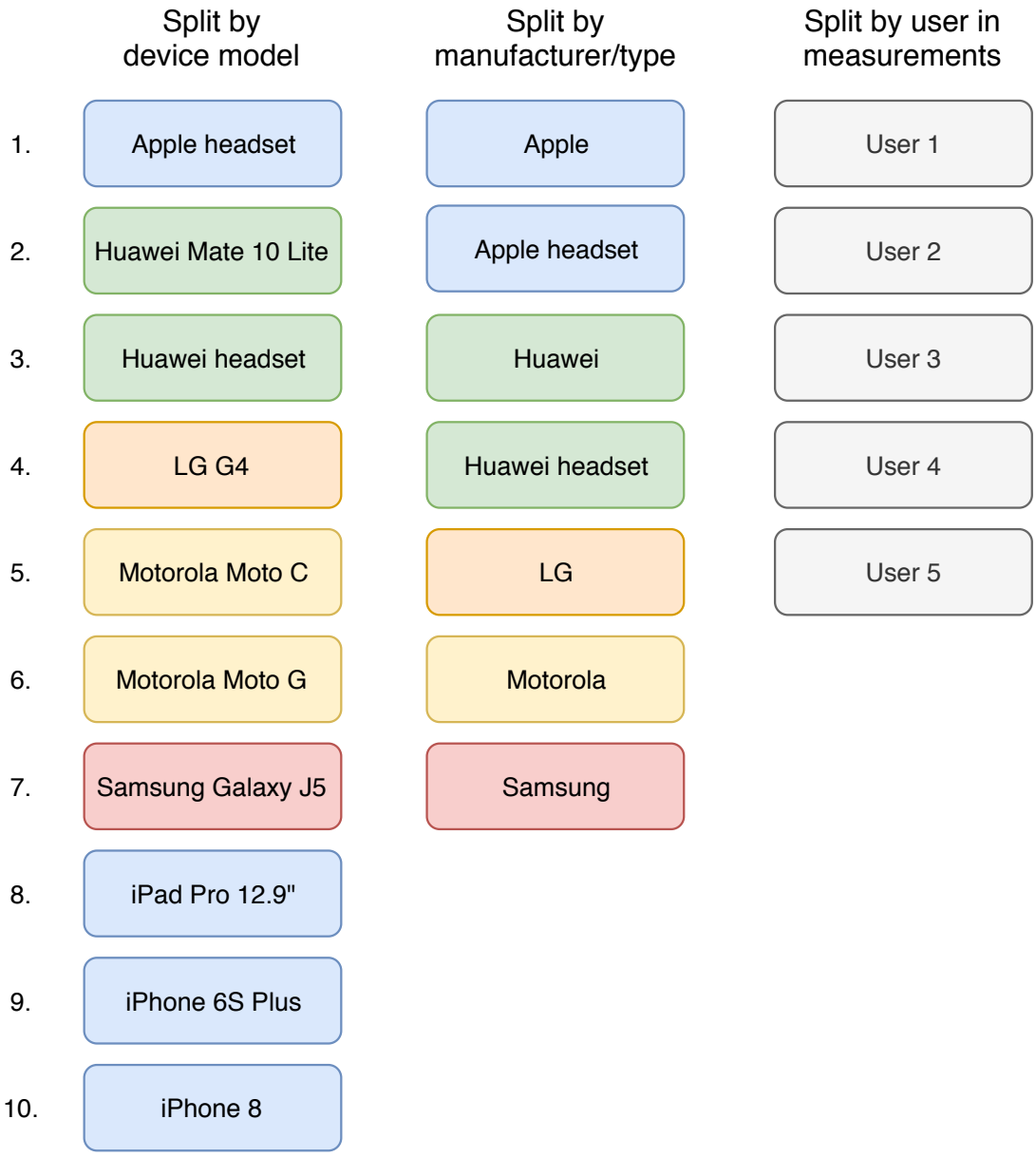


***Figure 4.4.*** *Partitioning of the device impulse responses.*

of the headsets individually and compare them with the phones. The manufacturer split creates in total seven partitions, the device model split ten[1] partitions, and the user split five partitions. Out of these partitioning methods, only the first two were used in the experiments.

### 4.3.2 Evaluation setup

Since there existed a cross-validation split for the word data to be classified, it was used as the basis. However, the point was not to evaluate the classifier itself but only the added

[1]There are 11 devices in the total list of devices, but the iPhone SE does not have the IRs measured with the person sitting at the table like the rest of the devices, so it was removed from this split.

gain of TADA, so the CV setup was not sufficient as is.

Due to TADA selecting the impulse responses and background noise samples randomly, there is some variation in the augmentation process in each consecutive run. If the number of speech data samples is small, there is also a chance that only a small subset of all the available impulse responses and background samples are used to augment the data. Therefore, running the same CV setup several times in a row without changing the augmentation setting would measure the consistency of the performance with the selected augmentation method and give a more reliable result for the added gain.

To combine the results of all five words, they had to be averaged. However, simply averaging the results of separate words would have resulted in big standard deviations because of the variation in performance between the words. To cancel the word differences, the results had to be first averaged run-wise across all words, and the final averages and standard deviations had to be calculated from these five run-average values. Because of these complications around the calculation of the standard deviation, they are only reported for the combined augmentation experiment.

In the experiments, when a subset is augmented, it means that it contains in addition to noisy data also clean data with a certain probability. The convolution steps were performed with a 30 % probability drawn from a uniform distribution. Because the SNRs are already randomly selected from a list in noise addition, an SNR of 96 dB corresponds practically to a clean sample. In the combined experiment there are therefore samples that are clean and samples that were processed with varying combinations of the three augmentation techniques.

Because of the large differences in the splits defined in the previous subsection, the setup is simplified to only using the room split for the RIRs in the first experiment and the device model split for the mobile device IRs in the second experiment. As mentioned before, only the third background noise dataset, which was split based on the user ids, is used for evaluation. In the fourth, combined experiment, the rooms are split by the measurement points and the mobile device IRs are split based on the manufacturers.

### 4.3.3 Experiment I: Exclusive rooms

In the first experiment, the effect of the convolutions with room impulse responses on the classifier performance was studied. In addition to measuring the gain from augmenting only the training data, robustness against new rooms was also tested. More specifically, robustness was tested by comparing the robustified classifier's performance between noisy test cases with seen and unseen distortions.

The first experiment consisted of four sub-experiments. First, all subsets were augmented with all except one room's impulse responses. Second, the training and validation sets were still augmented the same way but the test set was augmented with the room that

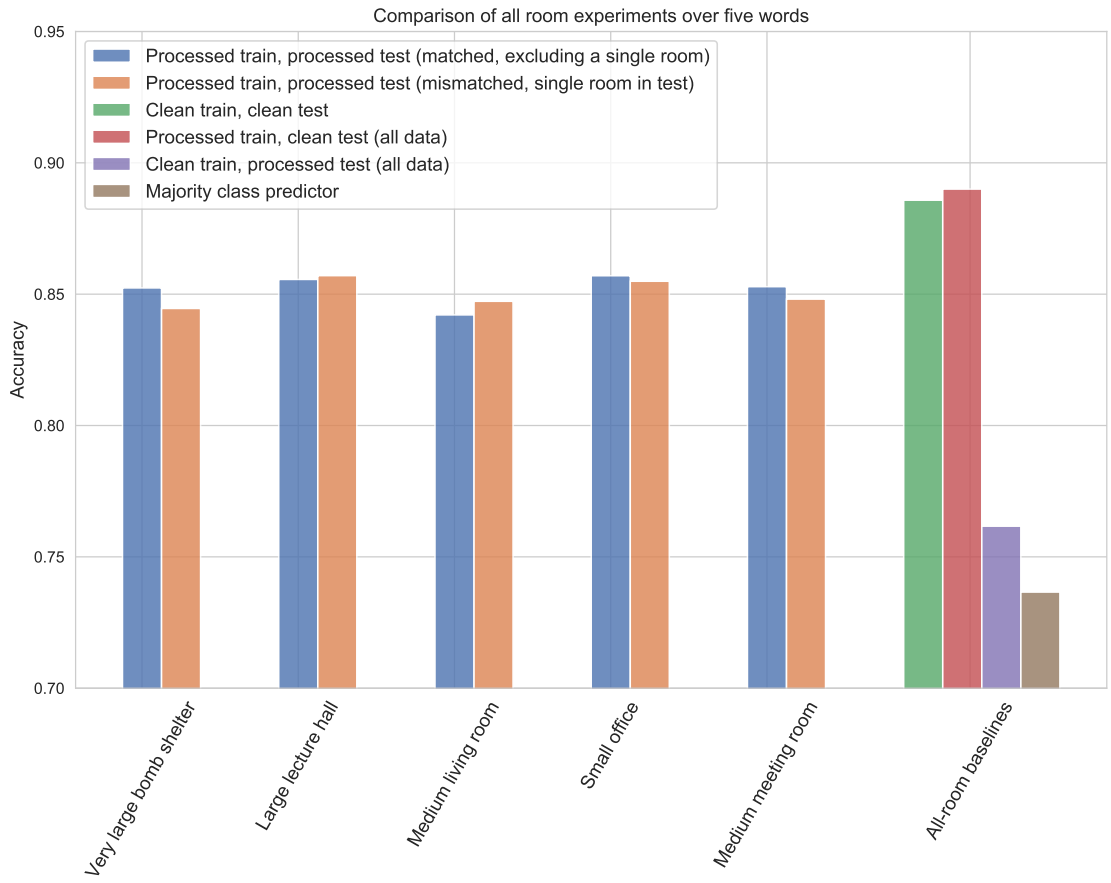


***Figure 4.5.*** *Room experiment results.*

was excluded in the first case. Third, the training and validation sets were augmented with all the room impulse responses and the test set was kept clean. Fourth, the training and validation sets were kept clean and the test set was augmented with all the room impulse responses. The system was also evaluated with all subsets clean (no augmentation) and with a zero rule (majority class) baseline classifier for comparison purposes. The results for the first experiment are shown in Figure 4.5. In the figure, the word *processed* is used when a subset is augmented with TADA.

The results for the first two sub-experiments (blue and orange bars) show that there is only a minor difference in the performance between the two cases. This suggests that the augmentation has made the classifier robust enough to handle unseen rooms without a significant drop in performance. The largest difference is expectedly observed with the bomb shelter alone in the test set.

The clean train, processed test case shows that TADA in fact makes the test set quite difficult to classify correctly when trained with clean data. However, augmenting all subsets with just room impulse responses can improve the accuracy of the classifier even above the clean-clean case.

### 4.3.4 Experiment II: Exclusive devices

In the second experiment, a similar setup as with the room impulse responses was used, this time only the impulse responses were from mobile device microphones. Again, the experiment was divided into four sub-experiments with the same two additional baselines for the nonaugmented case and the zero rule classifier. The results are shown in Figure 4.6.

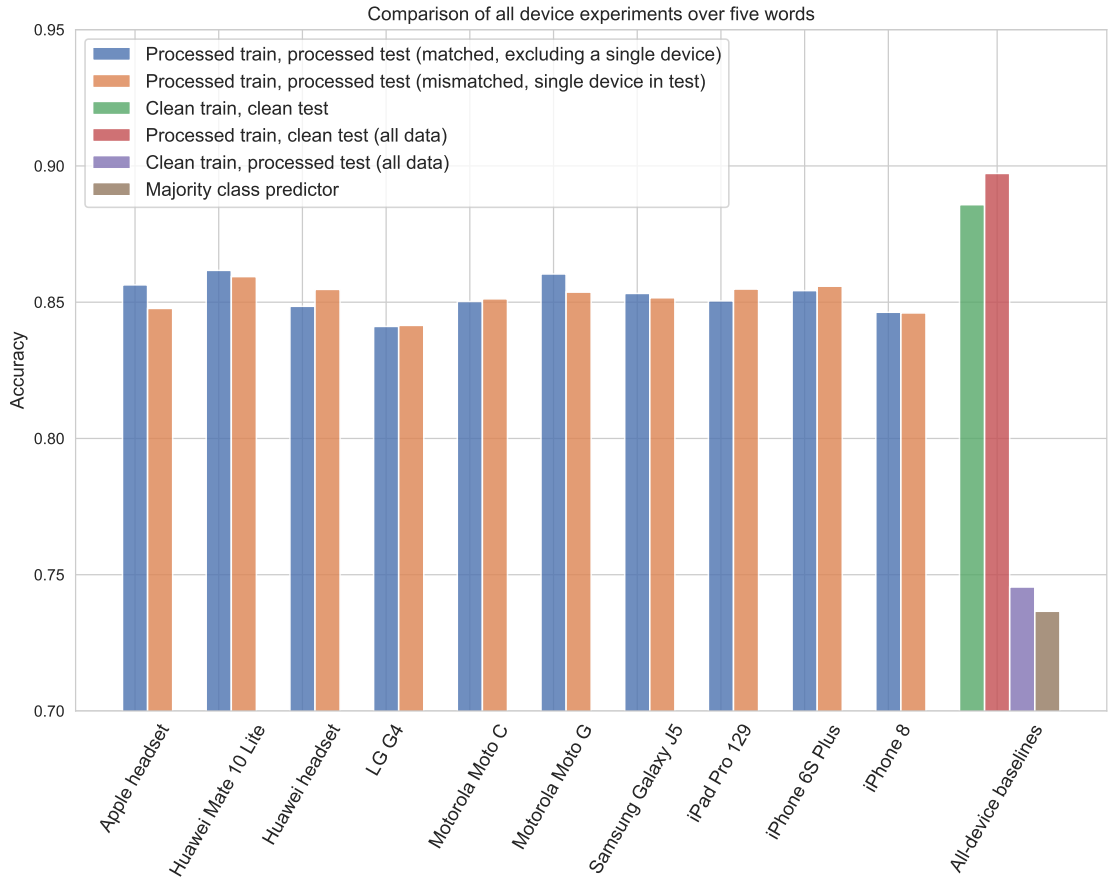


***Figure 4.6.*** *Device experiment results.*

The same behaviour as with the room experiment can be observed here: TADA is able to make the classifier robust against effects from new device microphones, the classifier performs better on clean test data when training data is augmented, and a classifier trained with clean data barely beats the zero rule classifier when tested against augmented test data. While the performance with clean training data and augmented test data is now lower than in the previous experiment, there is some improvement in the case where training data is augmented and the test data is kept clean. Overall, the levels of the mismatched and matched sub-experiments (blue and orange bars) vary around the 0.85 level as in the room experiment.

### 4.3.5 Experiment III: Varying SNRs

The third experiment consists of augmenting data with additive noise using acoustic scene backgrounds and the results are shown in Figure 4.7. Varying the SNRs adds another dimension to the augmentation process, and it was used in all of the noise sub-experiments.

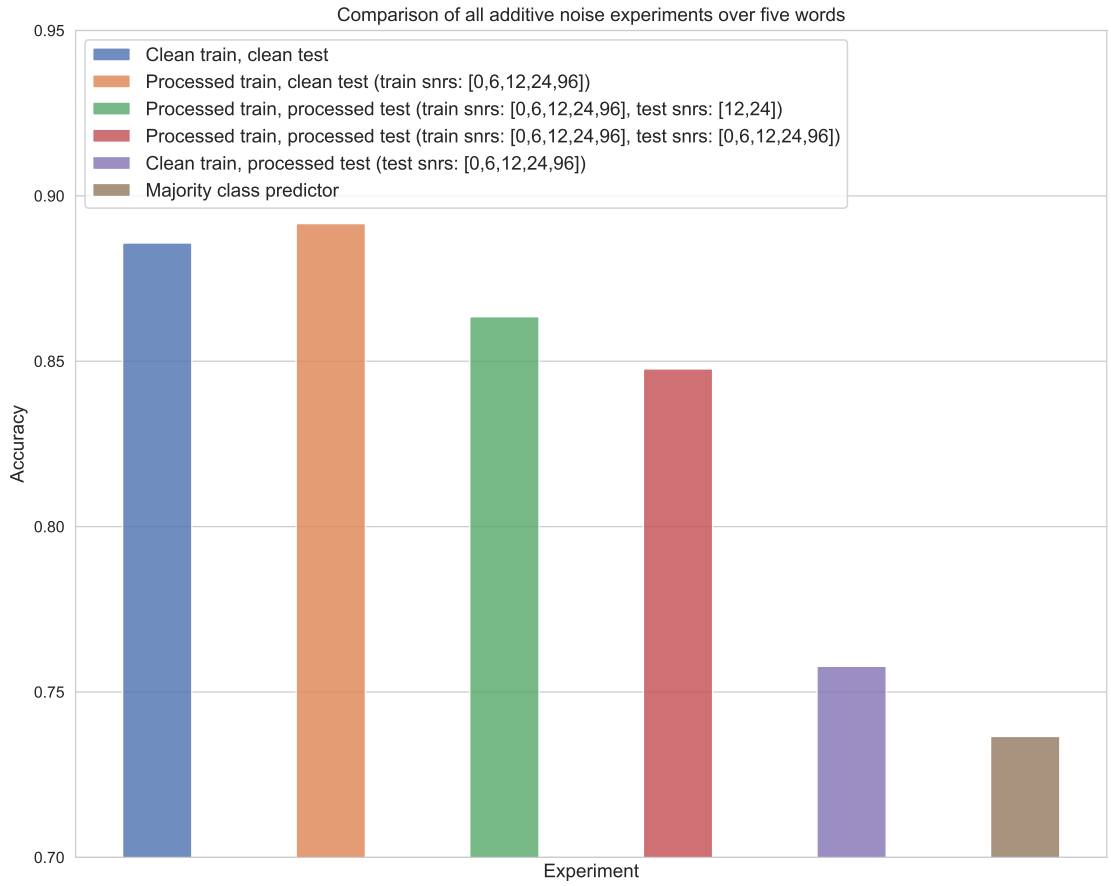


***Figure 4.7.*** *Additive noise experiment results.*

Four sub-experiments were carried out in the fashion of the impulse response experiments. Training with augmented data and testing with clean data, and the opposite case of training with clean data and testing with augmented data are comparable to the previous experiments. The same SNRs were used in the augmented subsets: 0 dB, 6 dB, 12 dB, 24 dB and 96 dB. In the other two sub-experiments (red and green bars) both the training and test data were augmented. In the first one, the SNRs in both subsets was kept the same, and in the second case the SNRs were higher by average (either 12 dB or 24 dB), therefore making the data less noisy. By doing this, the effect of the SNRs alone can be evaluated. In addition, correct behaviour of the system can be confirmed because the performance should improve when the SNRs stay higher making the test data more clean.

The classifier performs slightly better on clean test data when the training data is aug-

mented, although the difference is very minimal. Augmenting the test data with higher SNRs (less noisy) improves the accuracy over lower SNRs (more noisy), which proves that TADA is working correctly. Training with clean data and testing with augmented data yields an accuracy just above the zero rule also this time.

### 4.3.6 Experiment IV: Increasing level of augmentation

The fourth experiment combines all the implemented augmentation techniques into a three-step augmentation routine, which is the main idea of the system. An example split of the augmentation data used to augment the speech data in the first Monte Carlo run is shown in Figure 4.8. The room, background noise, and device partitions in test and validation sets are rotated for each run, while rest of the partitions are used for training.

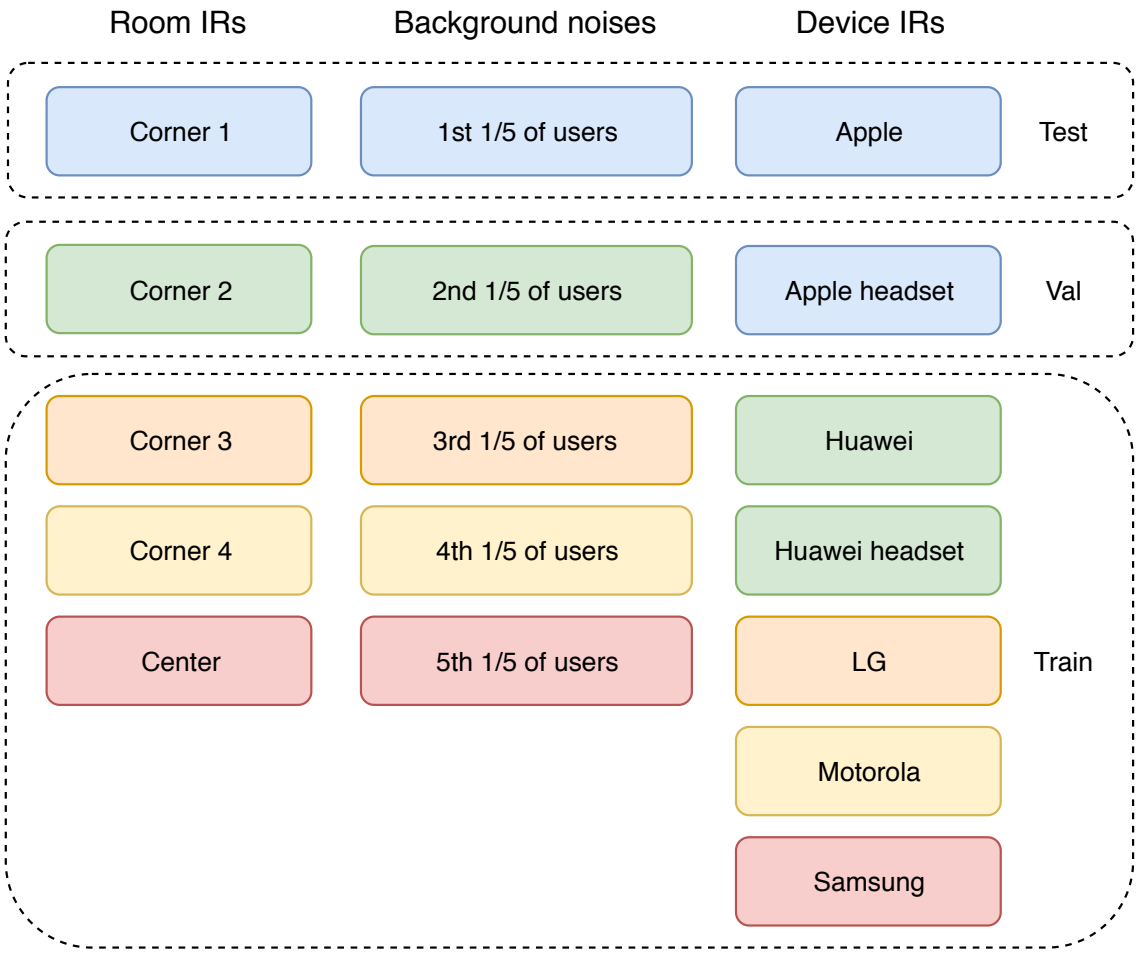


***Figure 4.8.*** *Partitioning of the augmentation data for the combined experiment.*

The results of the experiment are shown in Figure 4.9. The standard deviations only take into account the deviation between the five Monte Carlo runs. The augmentation starts with a room impulse response convolution, then background noise is added, and finally ends with a device impulse response convolution. The amount of augmented samples is controlled by the augmentation count variable.

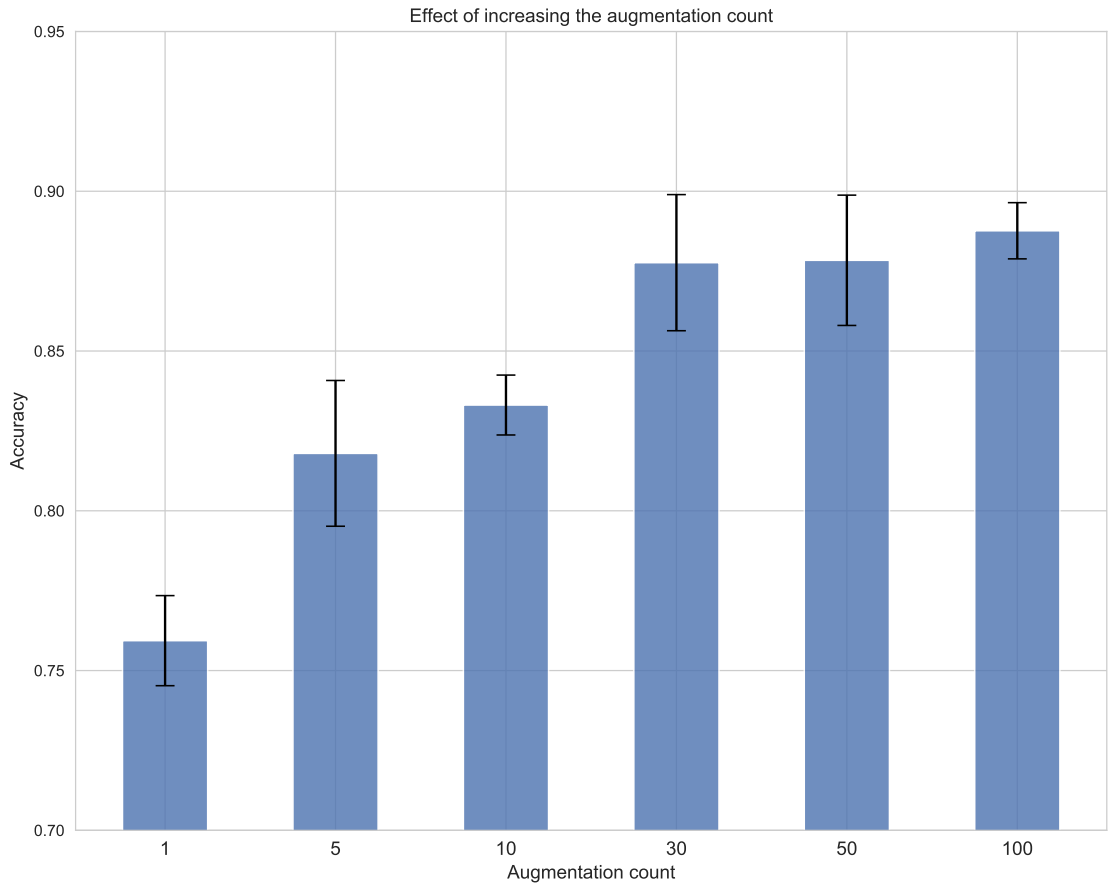


***Figure 4.9.*** *Augmentation count experiment results.*

On the horizontal axis, the augmentation count tells how many times the samples in the training data were processed with TADA. For example, when the augmentation count is one, all the training samples go once through TADA keeping the amount of data the same. When augmentation count is five, all the samples are processed five times, and the amount of data is multiplied up to five times the original. The test data was kept the same in all augmentation count experiments, and it was augmented with a constant augmentation count of 100 to make the task more difficult and to have room for accuracy improvement.

A gain of 12 % was achieved by augmenting the training data with the maximum augmentation count of 100. Accuracy starts to improve quickly when going from augmentation count of one to 30, but then it saturates. Augmentation count of 30 yields already almost the accuracy that is achieved with augmentation count of 100. Furthermore, the standard deviation closes the gap between the results with augmentation counts 30, 50 and 100. Therefore, it is not necessary to use higher augmentation counts when using TADA in this manner. Of course, the performance depends heavily on the test data, but nevertheless, this experiment shows the upper limit of performance gain received by using TADA with the three-step augmentation process.

# 5 CONCLUSIONS

In this work, distortions in acoustic systems and ways to minimize their effect were studied. Methods for improving the robustness of classification and detection systems were reviewed with emphasis on data augmentation techniques. Most common data augmentation techniques and their applicability for different audio analysis tasks were covered. Datasets for augmentation were reviewed and methods for collecting impulse responses were studied.

A system was proposed for augmenting audio data with a focus on audio analysis tasks operating on mobile device recorded audio. The augmentation simulates the process of recording audio in real environments with mobile devices. This simulation is achieved with a pipeline of convolution with a room impulse response, addition of an acoustic scene background, and convolution with a mobile device microphone impulse response.

The augmentation system was evaluated together with a pronunciation error classifier. Room effects, background noises, and device effects were studied separately. In each case, augmenting the data improved the performance of the classifier on both noisy and clean data. In addition, the combined three-step augmentation was found to improve the performance of the classifier until the data was augmented up to 30 times the original amount.

Although the performance of the classifier was shown to improve with the use of augmentation, there was no comparison of results received by other classification systems in the literature. The next step would be to test the augmentation system with more commonly used datasets to find out its applicability on a wider range of tasks. It could be worthwhile also to test different classifier architectures.

Additional study of combining the three techniques into pairs in multiple ways would allow the comparison of the effects of individual techniques better. Currently, it is only possible to conclude that each of the techniques alone is capable of making the classifier more robust.

Another possibility for future investigation would be the use of the augmentation system on the fly. In the current setup, data was created before the training, which restricts the amount of data to be augmented due to memory restrictions. Data generators can create data during the training process of a classifier, which would allow taking full advantage of the augmentation system by generating an endless amount of new data.

# REFERENCES


[1] A. F. Abka and H. F. Pardede. Speech recognition features: Comparison studies on robustness against environmental distortions. *2015 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*. 2015, 114–119.

[2] A. Acero. *Acoustical and Environmental Robustness in Automatic Speech Recognition*. Cambridge University Press, 1993.

[3] A. Acero, L. Deng, T. Kristjansson and J. Zhang. HMM adaptation using vector Taylor series for noisy speech recognition. *Sixth International Conference on Spoken Language Processing*. 2000.

[4] G. An. The Effects of Adding Noise During Backpropagation Training on a Generalization Performance. *Neural Computation* 8.3 (1996), 643–674.

[5] Apple Inc. *If the microphones on your iPhone, iPad, and iPod touch aren't working*. 2019. URL: https://support.apple.com/en-us/HT203792. (accessed: 04.09.2019).

[6] J. Benesty, J. Chen and Y. Huang. *Microphone Array Signal Processing*. Vol. 1. Springer Science & Business Media, 2008, 85–104.

[7] S. Bhardwaj. Audio Data Augmentation with respect to Musical Instrument Recognition. MA thesis. 2017. DOI: https://doi.org/10.5281/zenodo.1066137.

[8] X. Cui, V. Goel and B. Kingsbury. Data Augmentation for Deep Neural Network Acoustic Modeling. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 23.9 (2015), 1469–1477.

[9] L. Deng, A. Acero, L. Jiang, J. Droppo and X. Huang. High-performance robust speech recognition using stereo training data. *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*. Vol. 1. 2001, 301–304.

[10] R. F. Dickerson, E. Hoque, P. Asare, S. Nirjon and J. A. Stankovic. RESONATE: reverberation environment simulation for improved classification of speech models. *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. 2014, 107–117.

[11] A. Diment, E. Fagerlund, A. Benfield and T. Virtanen. Detection of Typical Pronunciation Errors in Non-native English Speech Using Convolutional Recurrent Neural Networks. *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. 2019.

[12] A. Diment, T. Virtanen, M. Parviainen, R. Zelov and A. Glasman. Noise-Robust Detection of Whispering in Telephone Calls Using Deep Neural Networks. *2016 24th European Signal Processing Conference (EUSIPCO)*. 2016.

[13] Earthworks, Inc. *M30 High Definition Measurement Microphone Datasheet*. URL: https://earthworksaudio.com/wp-content/uploads/2018/07/M30-Data-Sheet-2018.pdf. (accessed: 04.09.2019).

[14] Faber Acoustical, LLC. *iPhone 4 Audio and Frequency Response Limitations*. 2010. URL: https://blog.faberacoustical.com/wpblog/2010/ios/iphone/iphone-4-audio-and-frequency-response-limitations/. (accessed: 04.09.2019).

[15] F. J. Fahy. *Foundations of Engineering Acoustics*. Elsevier, 2000, 392–393.

[16] A. Farina. Simultaneous measurement of impulse response and distortion with a swept-sine technique. *Audio Engineering Society Convention 108*. 2000.

[17] J. L. Flanagan and R. Golden. Phase Vocoder. *Bell System Technical Journal* 45.9 (1966), 1493–1509.

[18] J. S. Garofolo. TIMIT Acoustic-Phonetic Continuous Speech Corpus. *Linguistic Data Consortium, 1993* (1993).

[19] S. W. Golomb et al. *Shift Register Sequences*. Aegean Park Press, 1967.

[20] Y. Gong. Speech recognition in noisy environments: A survey. *Speech Communication* 16.3 (1995), 261–291.

[21] Y. Han and K. Lee. Acoustic scene classification using convolutional neural network and multiple-width frequency-delta data augmentation. *CoRR* abs/1607.02383 (2016). URL: http://arxiv.org/abs/1607.02383.

[22] N. Jaitly and G. E. Hinton. Vocal Tract Length Perturbation (VTLP) improves speech recognition. *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*. Vol. 117. 2013.

[23] T. Ko, V. Peddinti, D. Povey and S. Khudanpur. Audio Augmentation for Speech Recognition. *Sixteenth Annual Conference of the International Speech Communication Association*. 2015.

[24] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer and S. Khudanpur. A study on data augmentation of reverberant speech for robust speech recognition. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017, 5220–5224.

[25] L. Lee and R. Rose. A Frequency Warping Approach to Speaker Normalization. *IEEE Transactions on Speech and Audio Processing* 6.1 (1998), 49–60.

[26] J. Li, L. Deng, R. Haeb-Umbach and Y. Gong. *Robust Automatic Speech Recognition: A Bridge to Practical Applications*. Academic Press, 2015, 43–46.

[27] D. Liang, Z. Huang and Z. C. Lipton. Learning Noise-Invariant Representations for Robust Speech Recognition. *2018 IEEE Spoken Language Technology Workshop (SLT)*. 2018, 56–63.

[28] A. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen and A. Y. Ng. Recurrent Neural Networks for Noise Reduction in Robust ASR. *INTERSPEECH*. 2012.

[29] B. McFee, E. J. Humphrey and J. P. Bello. A Software Framework for Musical Data Augmentation. *ISMIR*. 2015.

[30] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg and O. Nieto. librosa: Audio and Music Signal Analysis in Python. *Proceedings of the 14th Python in Science Conference*. 2015, 18–25.

[31] I. McLoughlin, H. Zhang, Z. Xie, Y. Song and W. Xiao. Robust Sound Event Classification Using Deep Neural Networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23.3 (2015), 540–552.

[32] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, W. Xiao and H. Phan. Continuous robust sound event classification using time-frequency features and deep learning. *PLoS One* 12.9 (2017).

[33] P. Mermelstein. Distance measures for speech recognition, psychological and instrumental. *Pattern Recognition and Artificial Intelligence* 116 (1976), 374–388.

[34] A. Mesaros, T. Heittola and T. Virtanen. *TUT Acoustic scenes 2017, Development dataset*. 2017. DOI: 10.5281/zenodo.400515. URL: https://doi.org/10.5281/zenodo.400515.

[35] J. Monge-Álvarez, C. Hoyos-Barceló, P. Lesso and P. Casaseca-de-la-Higuera. Robust Detection of Audio-Cough Events Using Local Hu Moments. *IEEE Journal of Biomedical and Health Informatics* 23.1 (2019), 184–196.

[36] P. J. Moreno. Speech Recognition in Noisy Environments. PhD thesis. 1996.

[37] V. Panayotov, G. Chen, D. Povey and S. Khudanpur. Librispeech: An ASR corpus based on public domain audio books. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, 5206–5210.

[38] G. Parascandolo, H. Huttunen and T. Virtanen. Recurrent Neural Networks for Polyphonic Sound Event Detection in Real Life Recordings. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016, 6440–6444.

[39] V. Pulkki and M. Karjalainen. *Communication Acoustics: An Introduction to Speech, Audio and Psychoacoustics*. John Wiley & Sons, 2015.

[40] A. Ragni, K. Knill, S. P. Rath and M. J. F. Gales. Data augmentation for low resource languages. *INTERSPEECH*. 2014.

[41] Z. W. Ras and A. Wieczorkowska. *Advances in Music Information Retrieval*. Vol. 274. Springer, 2010.

[42] S. Ravindran, D. V. Anderson and M. Slaney. Improving the noise-robustness of mel-frequency cepstral coefficients for speech processing. *SAPA-2006* (2006), 48–52.

[43] M. Ritter, M. Mueller, S. Stueker, F. Metze and A. Waibel. Training Deep Neural Networks for Reverberation Robust Speech Recognition. *Speech Communication; 12. ITG Symposium*. 2016, 1–5.

[44] C. Roads, A. Piccialli, G. D. Poli and S. T. Pope, eds. *Musical Signal Processing*. Swets & Zeitlinger, 1997.

[45] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Malaysia; Pearson Education Limited, 2016.

[46] J. Salamon and J. P. Bello. Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification. *IEEE Signal Processing Letters* 24.3 (2017), 279–283.

[47] J. Schlüter and T. Grill. Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks. *ISMIR*. 2015.

[48] J. J. Shynk. *Mathematical Foundations for Linear Circuits and Systems in Engineering*. John Wiley & Sons, 2016.

[49] D. Snyder, G. Chen and D. Povey. *MUSAN: A Music, Speech, and Noise Corpus*. arXiv:1510.08484v1. 2015.

[50] G.-B. Stan, J.-J. Embrechts and D. Archambeau. Comparison of different impulse response measurement techniques. *Journal of the Audio Engineering Society* 50.4 (2002), 249–262.

[51] S. S. Stevens, J. Volkmann and E. B. Newman. A Scale for the Measurement of the Psychological Magnitude Pitch. *The Journal of the Acoustical Society of America* 8.3 (1937), 185–190.

[52] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange and M. D. Plumbley. Detection and Classification of Acoustic Scenes and Events. *IEEE Transactions on Multimedia* 17.10 (2015), 1733–1746.

[53] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 2008, 569–572.

[54] T. Virtanen, M. D. Plumbley and D. Ellis. *Computational Analysis of Sound Scenes and Events*. Springer, 2018.

[55] T. Virtanen, R. Singh and B. Raj. *Techniques for Noise Robustness in Automatic Speech Recognition*. John Wiley & Sons, 2012.

[56] M. H. Weik. Nyquist theorem. *Computer Science and Communications Dictionary*. Springer US, 2001, 1127.

[57] D. Yu and L. Deng. *Automatic Speech Recognition*. Springer, 2016, 1–9.

[58] X. Zhao and D. Wang. Analyzing noise robustness of MFCC and GFCC features in speaker identification. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013, 7204–7208.