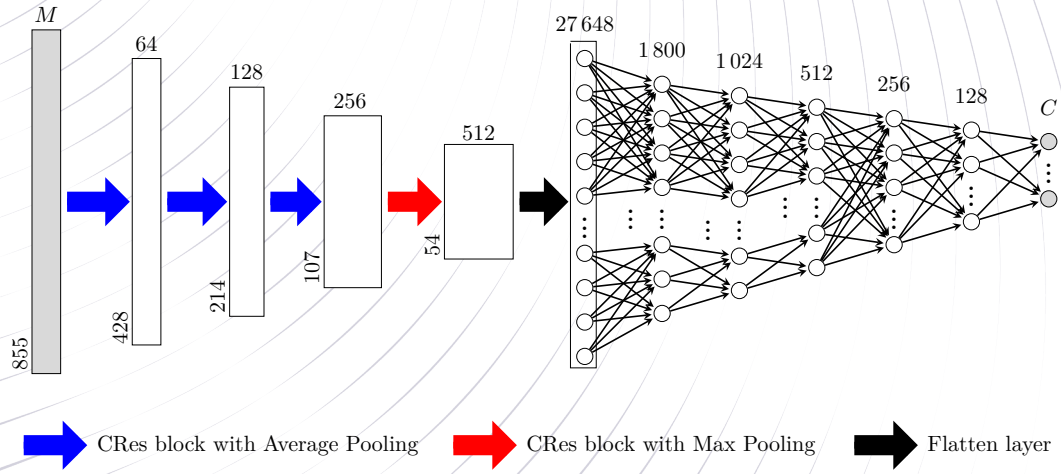
Classification of Room Characteristics Using Convolutional Neural Networks



Master Thesis
Morten Ø. Nielsen
Sanne D. Nielsen

Mathematical Engineering
Aalborg University
2019

AALBORG UNIVERSITY

STUDENT REPORT

Department of Electronic Systems
Department of Mathematical Sciences
Mathematical Engineering
Aalborg University
Fredrik Bajers Vej 7
9220 Aalborg
http://es.aau.dk
http://math.aau.dk

Title:
Classification of Room Characteristics Using Convolutional Neural Networks

Theme:
Machine Learning and Room Acoustics

Project Period:
Autumn 2018 and Spring 2019

Project Group:
MATTEK9-10 Gr. 5.213a

Participant(s):
Morten Ø. Nielsen
Sanne D. Nielsen

Supervisor(s):
Jan Østergaard
Henrik Grade (Autumn 2018)

Copies: 2

Page Numbers: 101

Date of Completion:
June 7th 2019

Abstract:

Classification of room characteristics based on Room Transfer Functions (RTFs) relates to various topics in acoustics, e.g. room response equalisation and forensic audio analysis.
In this Master's thesis, convolutional neural network models are proposed for room volume and wall admittance classification. The proposed models are trained using supervised learning based on simulated RTFs from a sound field model derived from the Helmholtz equation for wave propagation in small rooms. The classification performance of the models benefit from having microphones placed in structured grids compared to randomly placed microphones. For classification of the room volume, a classification accuracy above approximately 90% for more than one microphone is achieved, while for wall admittance classification the accuracy is above 80%. The models for room volume classification were more sensitive to additive noise than the wall admittance models, which still provided better then initial accuracy at 0dB SNR.

Danish Summary

Klassificering af rumkarakteristikker baseret på observationer af rumoverføringsfunktioner (RTF'er) relaterer sig til forskellige problematikker indenfor rumakustik, såsom rumresponsregulering og kriminaltekniske lydundersøgelser. Rumresponsregulering bruges bl.a. til højtalerer, hvor lydsignalet tilpasses efter størrelsen på rummet og dæmpning af væggene, med videre. For kriminaltekniske lydundersøgelser ønsker man at bestemme hvilket type rum et givet lydsignal er blevet opfanget i.

I dette kandidatspeciale foreslås to Convolutional Neural Network (CNN) modeller til klassificering af rummets volumen og væggenes admittanser. CNN modeller trænes via supervised learning til at klassificerer rumvolumen og vægadmittans ud fra log-magnitude responsen af simulerede RTF'er fra et given rum. De simulerede RTF'er er genereret ud fra en matematisk model af lydfeltet, der er udledt fra Helmholtz ligningen for udbredningen af lydbølger i små rum.

Klassificeringspræcisionen af modellerne undersøges for forskellige antal klasser og mikrofoner samt mikrofonernes placering i rummet, i forhold til hvilken indflydelse de har på præstationen. Derudover undersøges hvor sensitiv CNN modellerne er overfor additiv kompleks Gaussisk støj udtrykt i SNR.

Generelt kan der konstateres, at desto mere struktur, der er i placeringen af mikrofoner, desto bedre klassificeringspræcision kan der opnås. Antallet af tilgængelige mikrofoner havde en mindre betydning for de strukturerede placeret mikrofoner, men øgede præcisionen for tilfældig placerede mikrofoner. Overordet set kan der opnås en rumvolumeklassificeringspræcision på over 90% uafhængig af antal klasser for mere end én mikrofon, hvor en præcision på over 82% kan opnås for én specifikt placeret mikrofon. Ligeledes kan en vægadmittansklassificeringspræcision på over 80% generelt opnås uafhængig af antal klasser for mere end én mikrofon, hvor en præcision på over cirka 76% kan opnås for én specifikt placeret mikrofon.

CNN modellerne for klassificering af rumvolumen viste sig generelt at være mere sensitive overfor additiv kompleks Gaussisk støj end modellerne for vægadmittansklassificering. For 10dB SNR opnår CNN modellerne for klassificering af rumvolumen kun en klassificeringspræcision tilsvarende begyndelsespræcisionen af modellerne, når de

er trænet uden støj. Derimod opnår modellerne for vægadmittansklassificering en relative bedre klassificeringspræcision end begyndelsespræcisionen for helt ned til 0dB SNR. Når CNN modellerne er trænet med støj så observeres en generel forbedring af klassificeringspræcision, især for valideringsdata med tilsvarende SNR som træningsdata.

Preface

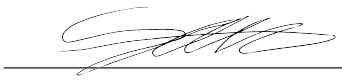
This Master Thesis (50 ETCS) is compiled by Morten Østergaard Nielsen and Sanne Damhus Nielsen, Master students in Mathematical Engineering at Department of Mathematical Sciences at Aalborg University. The IEEE-method is used for citations in the report. The scripts that are mentioned throughout the report are developed in the language Python 3.6 with Tensorflow API r1.12. A list and descriptions of the scripts can be found in Appendix A.

The report was written in cooperation with Bang & Olufsen A/S, and we would like to thank Martin Bo Møller, Pablo Martinez-Nuevo, and Sven Ewan Shepstone for guidance and helpful discussions.

As for prerequisites, the reader is expected to be familiar with vector spaces, the Fourier transform of continuous and discrete time signals, and general signal processing theory, statistics and understanding of the principals in machine learning. Furthermore, we will refer to the imaginary unit as it is done in engineering fields, thus j represents the imaginary unit.

Aalborg University, June 7$^{\text{th}}$ 2019

Morten Østergaard Nielsen
<mn14@student.aau.dk>



Sanne Damhus Nielsen
<saniel14@student.aau.dk>

Contents

Introduction

The sound experience and the atmosphere are some of the reasons for attending live concerts with your all time favourite band or artist. It is not uncommon that the sound perception can be terrible from where you are positioned in the concert hall. This is not necessarily the the fault of the band, but more likely some artefacts of the room acoustics of the concert hall. Depending on the size, architecture, and interior of a room, the sound behaves differently. Everyone can recall being in a relatively large room, e.g. an almost empty concrete hall, and being able to hear echoes of their own voice or footsteps. These echoes are caused by reflections of the sound. Whenever a sound wave hits a surface, e.g. a concrete wall, some of the sound wave will be passing through and some will be reflected. [15, Ch. 6 & Sec. 12.1]

The same phenomena also occurs in smaller rooms, such as a standard living room although the reflections may not be as audible as in larger rooms. This is not necessarily a bad thing, since it adds spaciousness and depth to the perception of the sound [4, Sec. 1]. On the other hand, it can introduce undesired acoustical artefacts [4, Sec. 1]. Some reflections create standing waves in the room, also referred to as resonances or room modes, which contribute to certain frequencies being amplified at different listener positions. Usually for smaller rooms, the room acoustics is dominated by the clearly separable resonances in the lower frequency range [15, Sec. 8].

In order to improve the listening experience, different methods and techniques have been proposed over the years (See Cecchi et al. [4]). In the literature, this is referred to by multiple names, one of them being room response equalisation [4, Sec. 1]. Modern day loudspeakers and loudspeaker-systems utilise different room response equalisation methods, and even some loudspeakers are capable of directing the sound as well and thereby creating so called “sweet spots”. A sweet spot is a specific point in the room, where the listening experience is close to perfect.

Creating a sweet spot requires information about the behaviour of the sound wave from the source to the listener position. This information can be obtained through the *Room Impulse Response (RIR)* or *Room Transfer Function (RTF)*. Either one of the RIR or the RTF has to be estimated through physical measurements of the sound

in the room. The problem here is that this requires a lot of microphones and if the smallest changes are added to the listening environment, e.g. moving the furniture around, the RIR and RTF changes too. Thus, the measurements must be repeated regularly, which means you would have to live with a constant microphone set-up in your living room.

Deep Neural Networks (DNNs) are well know for solving classification problems with high accuracy performance. An example here of is U-Net [23], which is a DNN for biomedical segmentation. However, in recent years DNNs have proven to be very powerful for generating synthetic data. Examples hereof are GAN [10] and PixelCNN [29, 30] for generating images, and WaveNet [28] for generating speech. Most of theses examples are so called *Convolutional Neural Networks (CNNs)*, which is a DNN that utilises a convolution operator. A solution to the problem of estimating the RTFs, for creating a sweet spot or room equalisation in general, could be an approach using CNN. Assuming that the RTFs from a few arbitrary positioned microphones and some position related information are available, it should be possible designing a CNN for generating a synthetic RTF of the sweet spot.

However, this could turn out to be a rather complicated problem and there is no guarantee for the chosen CNN architecture to be optimal. In this project, we will instead examine the classification of room acoustic parameters and characteristics. This is done as a first step towards a CNN for generating synthetic RTFs, since if a CNN is not capable of learning room characteristics, there is a very small probability for it being able to generate RTFs.

In order to get a better understanding of the components that the classification system consist of, the general system have been illustrated in Figure 1. First of all, the classification should be based on measured RTFs that are obtained from a set of microphones. However, we have decided to simulate and generate data using a sound field model to get a better control of the system parameters and a "infinite" supply of data. The RTFs are then preprocessed, which is customary for DNNs in general, before given as input to the CNN model. The model are trained using supervised learning, which means that the output of the CNN model are compared to the ground truth while training. The trained model are then used to classify the room acoustic parameters for new measured RTFs given in a validation set.

The report is organised as follows. First, the Problem Statement and delimitation of this project is presented. In Chapter 1, the motivation for the problem and a study of the state-of-the-art approaches in the literature are given, and we will state our contribution. An introduction to the necessary theory on acoustics and sound fields in enclosures are given in Chapter 2, followed by a description in Chapter 3 on how the theory is used for generating data sets and verifying the implementation. Chapter 4 introduces the necessary theory on DNNs, which are used for designing

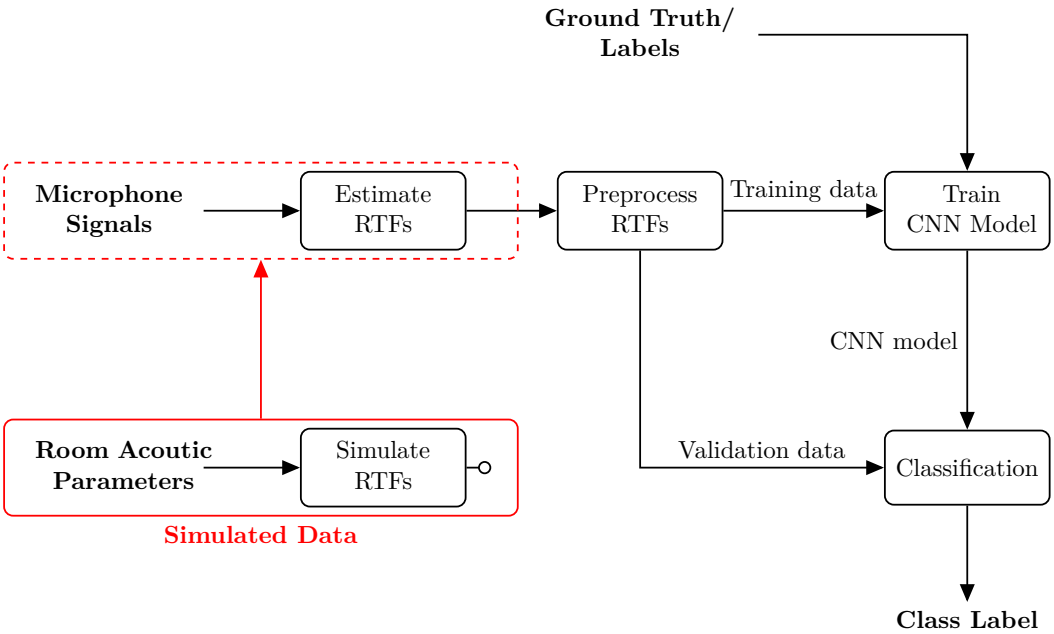


Figure 1: Block diagram of the general classification system. The red rectangles illustrate how the measured RTFs are replaced by simulations from a mathematical model.

the CNN models. The architecture of the CNN models is given in Chapter 5 along side a description of the conducted simulation experiments. In Chapter 6, we present the results of our experiments, and lead up to a discussion in Chapter 7. Finally, we end the report with a conclusion on the methods and results of this project.

Problem Statement

This project is made in collaboration with Bang & Olufsen A/S. Thus, the company have provided the following delimitations for the project. The rooms considered in this project are limited to ideal rooms, i.e. shoebox shaped with local reacting walls and uniform wall absorption. The simulated rooms follow the EBU Tech 3276 standard for listening rooms [6, Appendix 2] with the exception of the minimum volume for the rooms. The frequency range of the Room Transfer Functions (RTFs) considered in the project have been limited to the lower frequency region of 15 – 300 Hz.

Main Question:
How accurate can Convolutional Neural Networks (CNNs) classify the room characteristics; room volume and wall admittances, given RTFs for a set of microphones?

Sub Questions:

- How can realistic RTFs of ideal rooms, for training CNNs, be simulated from a mathematical model?
- What impact does the number and the placement of microphones have on the classification accuracy?
- Which impact does additive complex white noise, as a function of SNR, have on the classification accuracy?

Delimitations:
In order to simplify the classification problem the rooms are assumed time-independent, i.e. the room characteristics does not change over time. The room characteristics considered in this project are the room volume and wall admittances, since these two characterise the ideal room completely. Most of the other room characteristics, such as reverberation time, are related to the room volume and wall admittances. Thus it is sufficient to only examine the classification of these two characteristics.

The acoustic sources considered are assumed to be isometric point-like sources in order to simplify the sound field model used for generating data. This means that considerations regarding source directivity for example can be avoided.

Chapter 1

Background And Contribution

In this chapter, we examine the literature on room acoustics and machine learning topics related to the problem given above, and state our contribution as well. First, we will give a short description of the motivation for the problem and an overview of state-of-the-art approaches that have been studied previously in order to solve similar problems. We will end this chapter by introducing the simulation set-up considered in this project and give a system description of the proposed solution.

1.1 Motivation

The problem of classifying room acoustic parameters or characteristics, using only observations from the sound field, can be related to various problems in the different fields of acoustics. One example could be the field of *Room Response Equalisation (RRE)*[1], which aims at improving the sound reproduction in a room [4].

When a loudspeaker or loudspeaker-system emits a sound signal in a room, the sound signal will be reflected and attenuated by the walls [13, Ch. 2]. At certain frequencies, the reflections will create standing waves in the room, which amplify these frequencies in the sound signal [13, Ch. 3]. One can consider the room as a filter that is applied to the sound signal, which depends on the source and the listener position. Thus, the sound propagation from a source to a given listener position is given by the impulse response of the filter, which is referred to as the Room Impulse Response (RIR) in the literature [4].

RRE is applied to sound signals in a loudspeaker-system to counteract the undesired effects of the RIR, such as the effects from standing waves in the room, and thereby improve the listener experience [4]. RRE utilise advanced digital signal processing methods and techniques where most of them require estimates of the RIR or the

[1] Also referred to in the literature as "room equalisation", "room correction", "room dereverberation", "reverberation reduction", and many others [4, Sec. 1].

Room Transfer Function (RTF) [4].

In order to get a better understanding of how the classification problem relates to the field of RRE, we will explain the basic idea behind RRE. A simple mathematical interpretation of the received sound signal at a single listener position in the room is given by:

$$\tilde{s}(t) = (h * s)(t) + w(t) \tag{1.1a}$$

$$\tilde{S}(f) = H(f)S(f) + W(f) \tag{1.1b}$$

where $s, h, w : \mathbb{R} \mapsto \mathbb{R}$, $\tilde{s}(t)$ is the received sound signal, $s(t)$ is the sound signal emitted by a single loudspeaker in the room, $w(t)$ is some additive noise signal[2], and $h(t)$ is the RIR. Equation (1.1b) is obtained by taking the Fourier transform of (1.1a) where $H(f)$ is then referred to as the RTF. Henceforth, we will just refer to the RTF because of its relation to RIR through the Fourier transform. The idea behind RRE is to design a equalisation filter with transfer function $H_{eq}(f)$, which is applied to the sound signal before emitting it by a loudspeaker. This equalisation filter is designed such that the received sound signal at a listener position is approximately equal to original sound signal [4]:

$$\tilde{S}(f) = H(f)H_{eq}(f)S(f) \approx S(f)$$

where the noise term has been omitted. From the above, it seems fairly simple to set $H_{eq}(f) = H^{-1}(f)$, but this equalisation filter might not be stable[3], i.e. the output signal are unbounded and could be heavily amplified. Thus, there is no guarantee a stable inversion exists.

Various methods and techniques already exist to solve this problem, and we will not study these any further but refer the reader to Cecchi et al. [4] for an overview. However, most of these methods requires some estimate or model of the RTF, which in turn requires microphone measurements of the sound field or information about the room [4]. Most theoretical models of the RTF requires knowledge about the room characteristics such as the room dimensions, room volume, reverberation time, etc. [4] [13, Ch. 8]. An example hereof is the modal decomposition derived from the wave equation, which will be presented in the following chapter.

In order to do RRE in a given listening position estimation of the RTF is required. Estimating the RTF for any listing position requires measurements from multiple microphones placed in the room. The number of necessary microphones depends on the highest frequency representative in the RTF. This is related to the spatial sampling

[2] In practise this can be anything from a noisy ventilation system to a person speaking near by. Often, this will be simulated by a white noise process.

[3] Also referred to as Bounded-Input, Bounded-Output (BIBO) stability. The reader are referred to Oppenheim and Schafer [22] for more on the topic of general signal processing theory.

theorem (See Ajdler et al. [1, 2]), which is the equivalent of the Nyquist-Shannon theorem but for spacial fields. Hence, the number of necessary microphone could be immense for a small living room. Since the sound field changes with the room, e.g. if the furniture are rearranged, the microphone measurements must be repeated to make the RRE adaptive [4]. Thus, it would be beneficial to use a model of the sound field and use that for obtaining the RTFs instead.

The RTF heavily depends on the size of a room and the materials used on the walls, which vary from building to building. Thus, making a loudspeaker that is capable of doing RRE depending on the room will require the loudspeaker to at least distinguish between rooms and choose the appropriate method for the equalisation. Depending on which model of the sound field that are used for the equalisation, different room acoustical parameters are needed [4]. If these are not known in advance, they could be obtained from a few measurements of the sound field. This reduces to the classification problem under investigation in this project, but this is just one of the applications that the classification problem relates to.

In general, estimation or classification of room acoustic parameters can be useful in application of acoustics scene analyse in order to characterise the type of enclosure a given sound signal were received [26]. Such applications could be forensic audio analysis or music classification, e.g. was a sound signal received in a small living room or a great concert hall [26].

The problem in itself is also interesting from a theoretical point of view. As already mentioned, various models of the RTF based on room acoustic parameters exist in the field of acoustics. In room acoustic research, these models are compared to measurements of the RTF in order to study the efficiency of the models, i.e. how accurate they emulate the real RTFs [18]. But how accurately can the room acoustic parameters be estimated or classified from the measured RTFs instead? This is the motivation behind this project, to study how efficiently a Convolutional Neural Network (CNN) could classify room volume and wall admittances from RTF measurements.

Getting an idea of which acoustical parameters that can be learned by the CNN architecture helps with the design of a CNN architecture for generating synthetic RTFs in future work. Instead of only classifying room acoustic parameters and feed them into a model, the CNN architecture could learn the sound field model instead based on training data. Hence, the CNN could generate synthetic RTFs for RRE, only assuming that the RTFs from a few arbitrary positioned microphones and some position related information are available. For now, we examine the classification of room acoustic parameters.

1.2 State-Of-The-Art

Classification or estimation of room acoustic parameters have been studied in the literature of acoustics before, but different approaches have been utilised to solve the problem. In the following, we will study some of the state-of-the-art approaches in the literature, and give a short discussion on their performances and issues.

From an efficient estimate of the room volume, it can easily be inferred the room is large or small, e.g. a living room or concert hall. Various methods for estimating the room volume can be derived from the theory of sound fields in enclosures [18, 26, 27, 25]. Properties such as the model density or reverberation time (see Chapter 2) are related to the volume of a room, and can be obtained through the RIR or RTF.

Kuster [18] studied the reliability of four different methods for estimating the room volume from a single RIR. The four methods were applied to both simulated data from a mirror image source model, as well as measured RIRs from different rooms, in order to assess their performance [18]. The data set of measured RIRs includes measurements from standard listening rooms, lecture halls, concert halls etc., as well as meta data, e.g. number of receiver positions and absorption area [18]. The two of these four methods studied by Kuster [18] are the geometrical acoustics method and the diffuse field method. We refer the reader to the original paper by Kuster [18] for further details.

The geometrical acoustics method is derived from the temporal density of reflections, which is the total number of reflections over a given time interval when an impulse sound is emitted into the room. The method relies on the assumption that the reflections of a modelled RIR are represented by scaled Kronecker delta functions [18]. For measured RIRs, the reflections are obtained from the number of peaks [18]. Thus, the first step in this method is to apply an algorithm for finding the peaks in the magnitude of the RIR, in order to identify the reflections and obtain a binary signal [18]. By convolving this binary signal with a low pass moving average filter, an estimate of the temporal density can be obtained [18].

Since the theoretically temporal density of reflections increases with the time squared, Kuster [18] choose a minimum mean square estimator of the room volume, minimising the error between the true and estimated temporal density. For simulated RIRs from 2000 different rooms with volumes between $10\,m^3$ - $10\,000\,m^3$, the mean error between the true and estimated room volume was 3.8% and a standard deviation of 5.7% [18]. However for the measured RIRs, Kuster [18] reported that the method fails due to difficulties in identifying the reflections. This problem relates to the resolution limit in the time domain. In order to represent the reflections as Kronecker delta functions, a necessary condition is that the RIRs must have infinite band-

width, which is impossible in practise [18]. Because of this, two reflections arriving successively in time are not necessarily separable, and therefore, the method fails [18].

A similar method to the geometrical acoustics by Kuster [18] could be obtained in the frequency domain where the modal density (See Chapter 2) is considered instead. However, this approach still requires an algorithm for finding the peaks to identify the modes and does not solve the problem of resolution.

Regarding the other methods studied by Kuster [18], the diffuse field method was the one who provided the most consistent results with the exceptions of a few outliers. A diffuse field refers to that the sound pressure in a room is assumed to be the same everywhere. This is often obtained in relatively large rooms with no absorption at the walls [19, Sec. 2.5]. In a diffuse field, the mean square sound pressure in a room is related to the sound power of the signal, the room volume, and reverberation time. Reverberation time is a measure of the time, it takes for a sound signal to "die out" in a room after the source has been stopped emitting [19, Sec. 5.1].

For the diffuse field method, the average of the estimated room volumes are within $\pm 50\%$ of the true room volume when the outliers were excluded [18]. The average is computed over the number of receiver positions in the data sets for each room. The outliers are, according to Kuster's study, explained by measurement errors in the RIR or a particular room design feature, just to mention a few [18]. However, the diffuse field method suffers from a few drawbacks. One of them is that it relies on presence of initial time delay in the RIRs [18], which refers to the time delay from when a sound signal is emitted from the source, to when it is received at the microphone [18].

From the above, it seems that estimating the room volume for a single RIR is possible but with a large penalty in accuracy. On top of the paper by Kuster [18], Shabtai et al. [26, 27] proposed a *Gaussian Mixture Model (GMM)* based on the same acoustic features as Kuster [18], i.e. temporal density, reverberation time etc. The GMM was trained on different datasets for classifying a finite number of rooms using the room volume. For the first approach, the GMM was trained on a large set of RIRs as well as simulated RIRs using the image source method [26]. For the model trained with simulated RIRs, perfect classification was obtained for simulated RIRs, i.e. 100% accuracy, but the classification error increased for the observed RIR [26]. The increased classification error was caused by the model misclassifying rooms with similar room volume [26].

The following paper by Shabtai et al. [27] proposed an improved method, where the GMM was trained using reverberant speech signals. The reverberant speech signals were obtained from anechoic speech convolved with simulated and measured RIRs. Different methods were used to extract room volume features from the speech signals [27]. The best classification performance was obtained when room volume

features were extracted from speech signals that contained abrupt stops after single isolated words [27]. The classification error was relatively large for both simulated and measured RIRs, but worst for the measured RIRs. The GMM trained with anechoic speech convolved with simulated RIRs suffered from misclassifying rooms with similar room volumes, but also misclassified a few smaller rooms as larger rooms and vice versa [27]. The model trained with anechoic speech convolved with measured RIRs suffered from the same problem, but in this case the model classified the two largest rooms as the same room [27]. However, as a proof of concept Shabtai et al. [27] claimed that the method showed great potential and different modifications have been proposed for future work.

In the papers by Kuster [18] and Shabtai et al. [26, 27], they obtained estimates of the reverberation time using existing techniques based on integration of the RIRs. In a recent paper, Santos and Falk [25] proposed using a *Recurrent Neural Network (RNN)* to improve previous techniques based on the effects of reverberation on the modulation spectrum. The modulation spectrum is an acoustic signal representation obtained by taking the short-time Fourier transform of each subband envelope [25]. The proposed RNN model was an *Long-Short Time Memory (LSTM)* model, which was trained using reverberant speech signals [25]. Santos and Falk compared the LSTM approach with different benchmarks, such as a simple *Feed-forward Neural Network (FNN)* and a GMM, using root mean square error and mean absolute deviation as performance measures [25]. The proposed LSTM model showed a better performance than the other methods, which do not incorporate temporal dynamics of the modulation spectrum [25]. Also, the LSTM model turned out to be more robust to additive noise [25]. The aim of the paper by Santos and Falk [25] was to show a proof of concept, and thus, even higher performance of the LSTM model might be obtained by optimizing the hyper-parameters in future work.

In the papers by Shabtai et al. [26, 27], they showed a machine learning approach was possible for room classification, and compared to the work by Kuster [18], this approach seems more promising. Especially, since DNNs have shown great performance for classification problems, for instance U-Net [23] and the winner of the ImageNet competition 2012 AlexNet [17] [24]. Santos and Falk [25] proved that an LSTM model could be used for estimating the reverberation time of a room from reverberant speech. The reverberation time is related to the wall admittances, and in order to do wall admittance classification, a similar approach might be considered.

In the recent years, DNNs have shown great performance in generating both images and sound signal as well. E.g. PixelRNN and PixelCNN [29, 30] can generate high quality images, while WaveNet [28] have proven to synthesise high quality speech- and sound signals. We seek inspiration in some these DNN architectures, since they utilise some of the latest architectures for feature extraction in order to generate high quality signals.

As seen from the papers studied above, extraction of relevant features is necessary for classifying both room volume and reverberation time [18, 26, 27, 25]. The U-Net [23] architecture consists of two parts; a down-sampling and an up-sampling part. The down-sampling part consists of mainly convolutional layers, which are responsible for the feature extraction in U-Net. WaveNet [28] is based on a similar structure, but in order to increase performance, residual connections are utilised. Inspired by the feature extraction in U-Net [23] and WaveNet [28], a similar structure will be utilised in our contribution.

1.3 Contribution

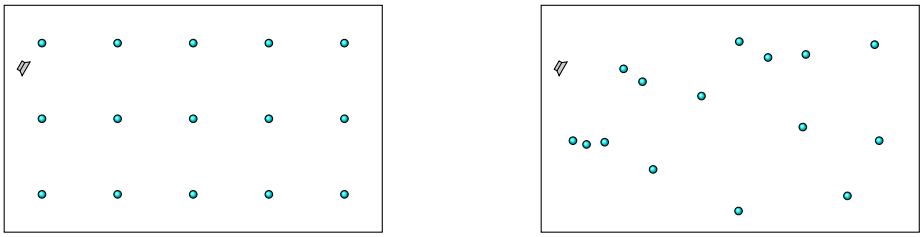
Inspired by the methods and techniques used in WaveNET, PixelCNN and U-Net [29, 28, 23], we intend to construct a CNN architecture that is capable of learning room acoustic characteristics, e.g. room volume and wall admittance. The CNN models will be trained using simulated RTFs based on a modal decomposition of the sound field (See Chapter 2). Our contributions are; **i)** designing a simulation environment for generating relevant and realistic data sets based on a modal decomposition of the sound field, and **ii)** designing a CNN architecture for classification of room acoustic parameters. In the following section, we will describe the simulation set-up used for generating the data sets and justify that the CNN models should be able to learn the room acoustic parameters from the RTFs. We will end this section with a preliminary analysis of the architecture of the CNN models.

1.3.1 Simulated Datasets

The training data, used for the CNN models in this report, is simulated RTFs for different positions in different rooms. The room shapes have been limited to shoe-box shaped rooms, e.g. rectangular rooms. Each surface of the room will have the same damping or absorption of the sound pressure, i.e. the wall admittance. The model presented in the following chapter is used for generating simulated RTFs for different room sizes and damping at arbitrary microphone and source positions in the room. The CNN models in this project will as input take several RTFs for different microphones positions and fixed source position. An illustration from above of different microphone configurations for a rectangular room are shown in Figure 1.1. Further details on how the datasets are generated will be given in Chapter 3.

1.3.2 Learnable Acoustic Parameters

A reasonable question to ask is, which room acoustic parameters are learnable for the CNN models examined in this project? The answer to this can be found by examining the generated RTFs ourselves and try to classify them. The argument is that if we can distinguish between a large and a small room simply by studying the visible changes in the RTFs, so will the CNN models. There is a wide range of room



(a) **(b)**

Figure 1.1: Illustration of the microphone configuration in a room seen from above. The cyan dots denote the microphones which are placed; (a) in a grid and (b) at random.

acoustic parameters, but many of these are related to each other (See Chapter 2 or [13, 19]). In this project, we have chosen to classify the room volume and wall admittances since these are the parameters necessary for generating RTFs with the given model in Chapter 2. Also, these acoustic parameters can be related to the damping coefficients, reverberation time, and the Schroeder frequency (See Chapter 2).

In the following, we will give a few examples to illustrate that it is possible to visibly distinguish between different room volumes and wall admittances. We will also justify the choice of a machine learning approach rather than a filter based method. The examples have been generate using the same model (See Chapter 2) as for generating the datasets. In order to examine the RTFs, we will study the magnitude response of the RTFs.

The magnitude responses for two different room sizes are illustrated in Figure 1.2. The two rooms are of size $5 \times 3 \times 2\,m^3$ and $9 \times 7 \times 5\,m^3$ respectively. The source is placed in one corner of the room and the microphone is placed in the opposite corner. As seen from the figure, the number of peaks in the magnitude response increase and becomes more dense for a large room. Hence, we can distinguish between the two RTFs by visible inspection, and thus the CNN models should be able to learn to classify the room volume as well.

The peaks in the magnitude response indicate the locations of resonance frequencies. Since these are related to the room size (See Chapter 2), one could argue that room classification could be done using filter-banks. E.g. one filter-bank could be used to locate the specific resonance frequencies for a room. Thus, by examining the output from the filter-banks, one could determine the room size and volume. However, the magnitude of the peaks change for different source and microphone positions in the room as illustrated in Figure 1.3 (See Chapter 3 for a description of the room coordinates). Thus, the filter-bank approach would be rather difficult since rooms might be misclassified if the measurements were obtained in arbitrary positions in the room. To avoid this issue, we choose CNN models, since they have been shown to be very robust for various classification problems (See [23] for an example).

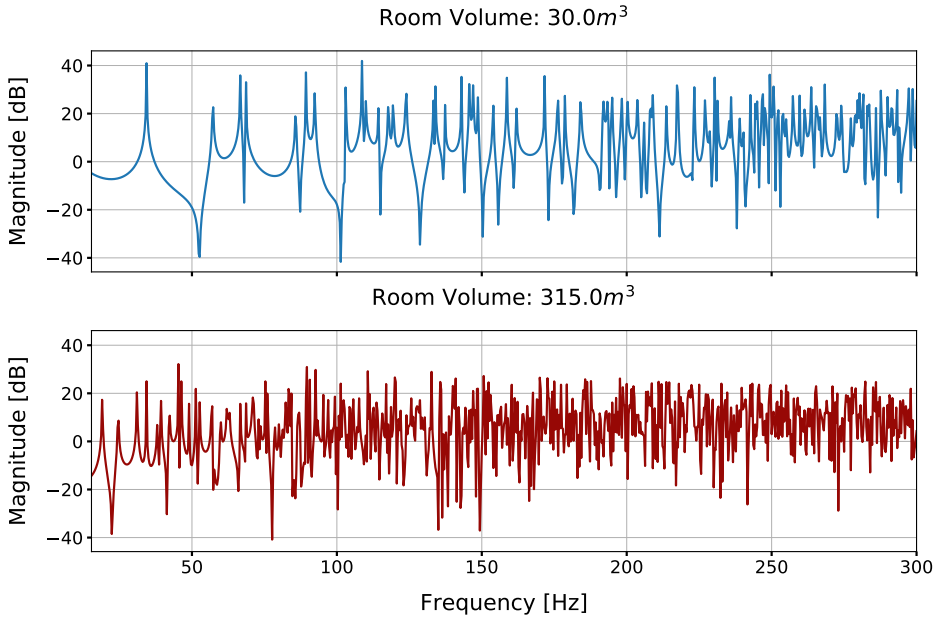


Figure 1.2: The magnitude response of an RTF for two different room sizes with volume; (**top**) $30m^3$ and (**bottom**) $315m^3$. The source and microphone are placed in opposite corners for both rooms.

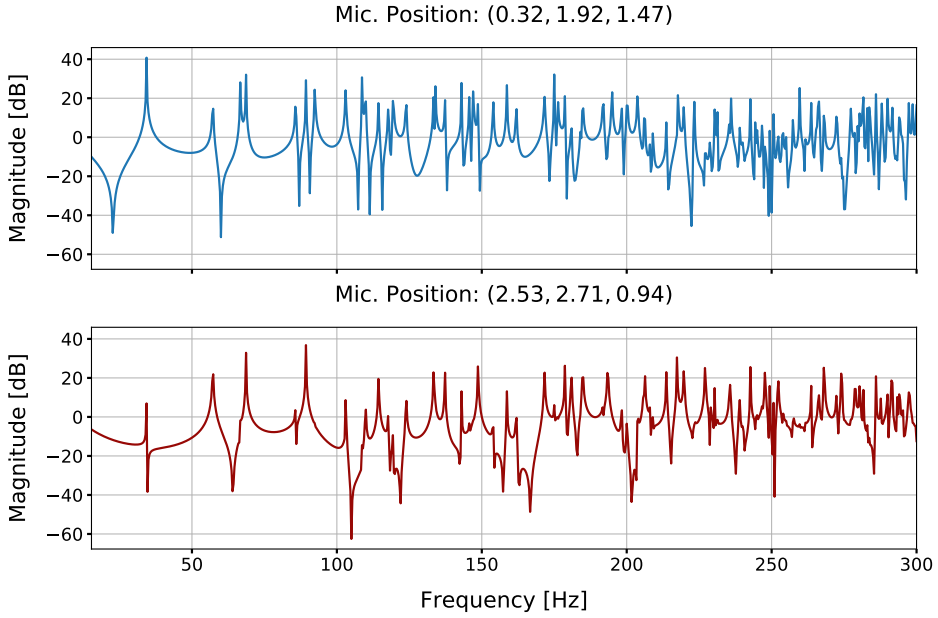


Figure 1.3: The magnitude response of an RTF for different microphone positions in the same room. The source is placed in the corner, $(0, 0, 0)$. The microphones are place in; (**top**) $(0.32, 1.92, 1.47)$ and (**bottom**) $(2.53, 2.71, 0.94)$.

By changing the wall admittances of the room, the magnitude response of the frequencies change as well. The magnitude response of the RTFs for different wall admittances are illustrated in Figure 1.4. The room is of size $5 \times 3 \times 2\,m^3$, and the source is placed in one corner of the room and the microphone in the opposite corner. As seen from the Figure 1.4, when the real component of the wall admittance is relatively small, the resonance frequencies introduce peaks in the magnitude response. As the real component of the admittance increases, the smoother the magnitude response becomes, since the resonance frequency gets more attenuated.

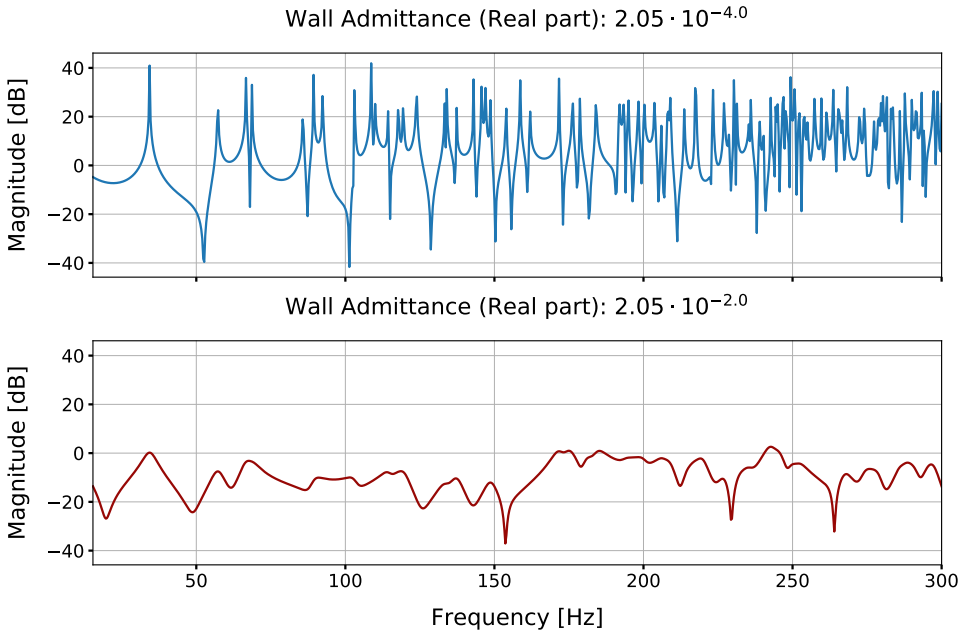


Figure 1.4: The magnitude response of an RTF for different real components of the wall admittances; (**top**) $2.05 \cdot 10^{-4}$ and (**bottom**) $2.05 \cdot 10^{-2}$. The same room size of $5 \times 3 \times 2\,m^3$ are used. The source and microphone are placed in opposite corners.

From the above examples, we argue that it should be possible for CNN models to classify the room volume and wall admittance from the RTFs.

1.3.3 CNN Models

The CNN models studied in this report will be designed with the same overall structure. In Figure 1.5, a generalized sketch of a CNN can be seen. The hidden layers have been depicted as a box in the figure, since the design of the hidden layers and their hyper-parameters are the parts of a CNN that can be modified in a attempt to improve the performance. The architecture of the hidden layers will be described later in Chapter 4. However, since the application of the CNNs is known, we can analyse the input and output layer. The design of the input and output layer will

be described separately in the following. The training of the CNNs will be based on supervised learning, since it is easier to guide the output towards a specific goal and verify it [5, p. 274].

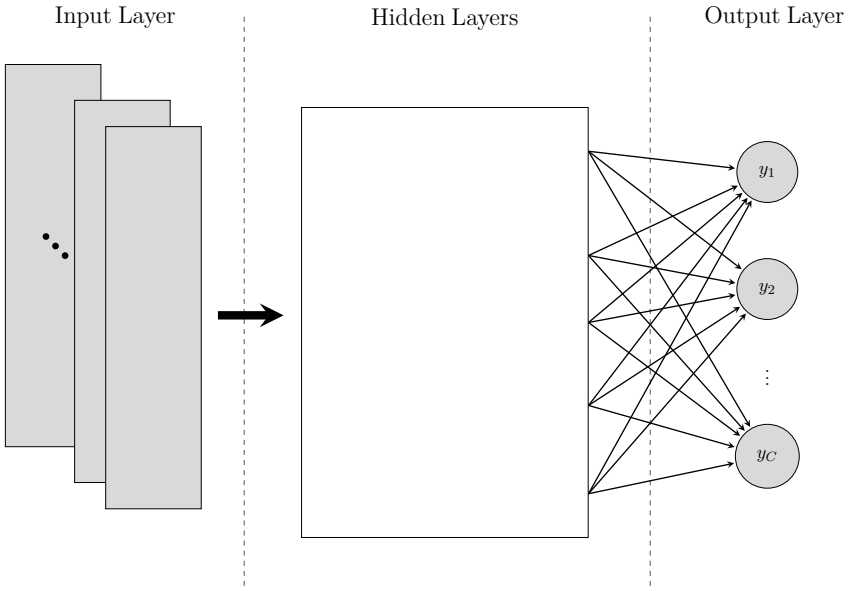


Figure 1.5: A CNN consisting of an input-layer, an output-layer and hidden layers. The architecture of the hidden layers have been depicted as an white box.

Input-layer

The input of the CNN models are two-dimensional, i.e. $\mathbb{R}^{F\times M}$, where F is the number of frequency samples and M is the number of microphones. The number of frequency samples is computed from the sampling rate, f_s, and the excitation time, t_s, used for the simulation. The excitation time refers to how long time the microphones are "recording". In this project, the frequency axis has been limited to $[15, 300]$ Hz, thus the number of frequency samples is given by:

$$\begin{aligned} L :&= \left\lfloor \frac{f_s t_s}{2} \right\rfloor - 15 t_s \\ &= \left\lfloor \frac{(f_s - 30) t_s}{2} \right\rfloor \end{aligned}$$

E.g. a sampling rate of 600Hz and excitation time 3s gives $F = 855$ frequency samples. However, in order to feed the data to the CNN models the data must be transformed into real values, because CNNs, and ANNs in general, does not take complex valued inputs and RTFs are indeed complex due to the Fourier-transform (See Chapter 2). There is multiple transformations that can solve this problem. One way would simply be to divide the RTFs into two; a real-part and a imaginary-part.

Another way would be to use the magnitude response and/or the phase response of the RTFs. For this project, we have decide to use the magnitude response only and will not go further into the other methods.

Output-layer

The output-layer of a CNN model is a so called dense-layer (See Chapter 4). The number of nodes in the layer, thus the dimension of the output, is determined by the number of classes C (See Figure 1.5). There is no definite choice on the number of classes, since this is one of the parameters that will be examined in the project. The interpretation of the output is a discrete probability distribution, which describes the probability of the input belonging to each of the classes. Thus, $y_i \in [0, 1]$ and the output entries most sum to 1, i.e. $\sum_{i=1}^{C} y_i = 1$. One can determine the most probable class $C^\star$ by:

$$C^\star = \arg\max_{i \in \mathcal{C}} y_i$$

where $\mathcal{C} := \{1, 2, \ldots, C\}$. Since the CNN models are trained using supervised learning, the prediction $C^\star$ is compared to the true label. By counting the number of correct class matches, the classification accuracy can be obtained.

In this chapter, a literature study were given and our contribution to classification of room acoustic characteristics were presented. Thereby, we are ready to give a short introduction to the theory of room acoustics and the sound field model in the next chapter.

Chapter 2

Sound Fields in Enclosure

The purpose of this chapter is to give an introduction to the necessary acoustic- and sound field theory for enclosures, which will be used for generating data sets and verify the implementation. Note that this is only a brief introduction without further derivation of the stated results.

The perceived sound quality of a room is largely dependent on the room characteristics. For given positions, the room characteristics can be captured by measuring the Room Transfer Function (RTF). The RTF changes depending on the position and distance to the source due to reflections from the surfaces of the emitted waves [4, Sec. 2]. We aim to classify room characteristics using convolutional neural network models based on different measurements of the RTFs in a room and this lead us to derive a model for simulating these measurements.

When emitting a sound wave in a medium, its molecules will be set in motion by the perturbation from the wave. These molecules will then oscillate back and forth in the direction of propagation. This introduced displacement of molecules changes the pressure and is known as the *sound pressure*. [13, Sec. 2.1]

The mathematical description of the changes in pressure produced by a sound wave is given by the *wave equation*, which can be derived from a series of basic physical laws and relations. These are conservation of mass, conservation of momentum and that sound induces nearly no flow or heat (See [13, Sec. 2.2] for the construction of the wave equation). The homogeneous wave equation is given as the linear second order *Partial Differential Equation (PDE)* [13, Sec. 2.2] [16, Ch. 3]:

$$\Delta p = \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} \tag{2.1}$$

where Δ is the Laplacian operator [16, Sec. 1.2] and c is the propagation speed ($c \approx 343.216[m/s]$ in air at 20°C [13, Sec. 2.2]). The function $p : \Omega \times \mathbb{R}_+ \mapsto \mathbb{R}$ describes the sound pressure on some open set $\Omega \subset \mathbb{R}^n$. Note, that the equality only

holds under the given assumptions as stated in [13, Sec. 2.2].

Since the molecules in the given medium are oscillating in the direction of propagation, the sound pressure p is expected to change harmonically [13, Sec. 3.1]. This means that the Fourier transformation can be utilised to get a better understanding of the sound pressure. Note that by doing so we implicitly assume the emitted sound signals to be in the L^2-space and twice continuous differentiable, in order to solve the PDE, i.e. in C^2. Let $\hat{p}(\boldsymbol{r}, \omega) := \mathcal{F}_t\{p(\boldsymbol{r}, t)\}(\omega)$ denote the Fourier-transformed sound pressure p w.r.t. time t at position $\boldsymbol{r}$. Then by taking the Fourier transform of the wave equation (2.1) and rearranging the terms, we obtain the PDE:

$$\Delta\hat{p}(\boldsymbol{r}, \omega) + k^2\hat{p}(\boldsymbol{r}, \omega) = 0 \tag{2.2}$$

where $k := \omega/c$ denotes the *wave number*. This PDE is known as the *homogeneous Helmholtz equation* and describes wave propagation in free-field without any sources present [31, Sec. 2.4]. In order to introduce sources into the equation, one obtains the inhomogeneous Helmholtz equation:

$$\Delta\hat{p}(\boldsymbol{r}, \omega) + k^2\hat{p}(\boldsymbol{r}, \omega) = -j\omega\rho Q(\boldsymbol{r}, \omega), \tag{2.3}$$

where ρ is the density of the medium and $Q(\boldsymbol{r}, \omega)$ is a distribution of sources that belongs to the domain Ω [13, Sec. 7.6]. From this point and onward, we assume that only a single source is present and that it is a isometric point-like source (See [13, Ch. 9]). Thus, $-j\omega\rho Q(\boldsymbol{r}, \omega) = -j\omega\rho Q\delta(\boldsymbol{r} - \boldsymbol{r}_0)$, where $\boldsymbol{r}_0 \in \Omega$ is the source position and $Q \in \mathbb{C}$ must fulfil $j\omega\rho Q = 1$ [13, Sec. 7.6].

One way to find a solution for the inhomogeneous Helmholtz equation (2.3) is by considering a solution G_ω for:

$$(\Delta + k^2)G_\omega(\boldsymbol{r}, \boldsymbol{r}_0) = -\delta(\boldsymbol{r} - \boldsymbol{r}_0), \tag{2.4}$$

where G_ω is called a Green's function [7, Ch. 10]. Thus, one can obtain a solution for (2.3) from:

$$\hat{p}(\boldsymbol{r}, \omega) = \int_\Omega Q(\boldsymbol{r}_0, \omega)G_\omega(\boldsymbol{r}, \boldsymbol{r}_0)\,\mathrm{d}\boldsymbol{r}_0 \tag{2.5}$$

However, in order to obtain a solution G_ω as described above, one must describe some boundary conditions for the problem. These boundary conditions can be obtained from the room characteristics, and together with the PDE (2.3) one have to solve a *boundary value problem (BVP)*. An important property of the Green's function is that it can be considered the RTF for the room of interest, if $j\omega\rho Q = 1$ [13, Sec. 7.6]. Hence, we wish to derive the Green's function for the rectangular room in order to obtain a model for our simulation later.

2.1 The Rectangular Room

In order to find a suitable Green's function, and thereby obtain a model for the sound field in a room, we state a BVP. In the following, the boundary conditions for a rectangular room will be derived and the BVP will be stated. Also, the Green's function for two cases will be stated as well. The following is based on a combination of [13, Ch. 8], [19, Ch. 3], and [12, Sec. 9.2].

The rooms that are considered in this project are rectangular with the dimensions $l_x \times l_y \times l_z$, where $l_x, l_y, l_z \in \mathbb{R}$. Define $\mathcal{B} := (0, l_x) \times (0, l_y) \times (0, l_z)$ as the interior of the room. Then, let $\partial\mathcal{B}$ denote the boundaries of $\mathcal{B}$, i.e. the walls of the room. In the following, we will consider two cases: rigid- and non-rigid walls. Rigid walls refers to that the admittance of the walls are zero, i.e. no damping of the reflections, and non-rigid walls refers to the contrary.

2.1.1 Boundary Value Problem

In a room with non-rigid walls, the sound reflections from the walls will be attenuated because the walls will "absorb" some of the energy from the emitted sound signal [13, Ch. 2]. How much of the intensity of the sound signal that is absorbed and reflected is determined by the admittance of the wall [13, Ch. 2]. The wall admittance is defined as the reciprocal of the impedance. We will in the following use the wall admittance rather than the impedance, since it becomes convenient later. We assume that the room under consideration has locally reacting walls [19, Sec. 3.1], and that all of the walls have the same constant admittance, $\beta \in \mathbb{C}$.

The boundary conditions for the room of interest can be derived from one of the basic conservation laws; *conservation of momentum* [19, Sec. 1.1]. This is one of the physical laws that can be used to derive the wave-equation (2.1). Conservation of momentum states:

$$\nabla p = -\rho_0 \partial_t \boldsymbol{v} \tag{2.6}$$

where $\boldsymbol{v}$ is the vector particle velocity and ρ_0 the equilibrium density [19, Sec. 1.1]. By taking the Fourier transform with respect to the time component, one obtains:

$$\nabla \hat{p} = -j\omega\rho_0 \hat{\boldsymbol{v}} \tag{2.7}$$

The velocity component normal to any boundary, i.e. wall, can then be obtained as:

$$\hat{v}_n = -\frac{1}{j\omega\rho_0} \nabla \hat{p} \cdot \boldsymbol{n} = \frac{j}{\omega\rho_0} \frac{\partial \hat{p}}{\partial n} \tag{2.8}$$

where $\boldsymbol{n}$ is the outward normal vector to any wall and $\frac{\partial}{\partial n}$ denotes the partial derivative in the direction of $\boldsymbol{n}$. The velocity, $\hat{v}_n$, can be replaced by $\frac{\hat{p}\beta}{\rho_0 c}$ to obtain [19,

Sec. 3.1]:

$$\frac{\partial \hat{p}}{\partial n} = -j\frac{\omega}{c}\beta\hat{p} = -jk\beta\hat{p} \tag{2.9}$$

These boundary conditions are so called *Neumann boundary conditions* [16, Sec. 1.4]. Together with the inhomogeneous Helmholtz equation (2.3), we obtain the BVP for the rectangular room:

$$\begin{cases} \Delta\hat{p}(\boldsymbol{r}) + k^2\hat{p}(\boldsymbol{r}) = -j\omega\rho Q\delta\left(\boldsymbol{r}-\boldsymbol{r}_0\right) & \text{in } \mathcal{B}, \\ \dfrac{\partial \hat{p}}{\partial n} = -jk\beta\hat{p} & \text{on } \partial\mathcal{B} \end{cases} \tag{2.10}$$

where the admittance $\beta = 0$ if the walls are rigid and $\beta \neq 0$ if the walls are non-rigid.

2.1.2 Green's Functions

In the previous, we derived the BVP for the rectangular room. In the following, the Green's functions for rigid- and non-rigid walls will be stated. A thorough derivation of the Green's function for the case with rigid walls is given Appendix B and we refer to [13, Ch. 8], [19, Ch. 3], and [12, Ch.9] for the details on the derivation for non-rigid walls.

First, consider the case with rigid walls, i.e. $\beta = 0$ and $j\omega\rho Q = 1$. Thus, the BVP becomes:

$$\begin{cases} \Delta\hat{p}(\boldsymbol{r}) + k^2\hat{p}(\boldsymbol{r}) = -\delta\left(\boldsymbol{r}-\boldsymbol{r}_0\right) & \text{in } \mathcal{B}, \\ \dfrac{\partial \hat{p}}{\partial n} = 0 & \text{on } \partial\mathcal{B} \end{cases} \tag{2.11}$$

The Green's function for this BVP is given as (See Appendix B):

$$G_\omega(\boldsymbol{r},\boldsymbol{r}_0) = -\frac{1}{V}\sum_{|N|=0}^{\infty}\frac{\psi_N(\boldsymbol{r})\psi_N(\boldsymbol{r}_0)}{k^2-k_N^2}, \quad \text{where} \tag{2.12}$$

$$N := (n_x, n_y, n_z) \in \mathbb{N}_0^3, \quad |N| = n_x + n_y + n_y, \quad V = |\mathcal{B}|,$$

$$k_N^2 = \left(\frac{n_x\pi}{l_x}\right)^2 + \left(\frac{n_y\pi}{l_y}\right)^2 + \left(\frac{n_z\pi}{l_z}\right)^2, \quad k = \frac{\omega}{c} = \frac{2\pi f}{c}$$

$$\psi_N(\boldsymbol{r}) = \sqrt{\varepsilon_{n_x}\varepsilon_{n_y}\varepsilon_{n_z}}\cos\left(\frac{n_x\pi}{l_x}x\right)\cos\left(\frac{n_y\pi}{l_y}y\right)\cos\left(\frac{n_z\pi}{l_z}z\right),$$

$$\varepsilon_i = \begin{cases} 1 & \text{for } i = 0 \\ 2 & \text{for } i \in \mathbb{N} \end{cases}$$

$\boldsymbol{r} = [x\,y\,z]^\intercal \in \mathcal{B}$ and $\boldsymbol{r}_0 \in \mathcal{B}$ is the position of the source inside the room. The functions ψ_N make up an orthogonal basis in $L^2(\mathcal{B})$ for a fixed ω and are called the *modes* [13, Sec. 8.1]. Also, (k_N^2, ψ_N) are eigenpairs to $-\Delta$ with homogeneous Neumann boundary conditions, and we will return to the relation between k_N and

ψ_N later. An important property of the Green's function in (2.12) is that, it is symmetric in the sense that the microphone and source position can be interchange, i.e. $G_\omega(\boldsymbol{r}, \boldsymbol{r}_0) = G_\omega(\boldsymbol{r}_0, \boldsymbol{r})$.

Now consider the case where $\beta \neq 0$. The BVP in (2.10) becomes complicated to solve and hard to obtain a Green's function as well. It can be shown that the Green's function is in the same form as (2.12), but with a different set of eigenfunctions, ψ_N, and with complex eigenvalues, k_N [12, Sec. 9.2]. However, under the assumption that the wall admittance is small, i.e. $|\beta| \ll 1$, then the Green's function is similar to (2.12) with an added imaginary term in the denominator [19, Sec. 3.3]:

$$G_\omega(\boldsymbol{r}, \boldsymbol{r}_0) = -\frac{1}{V} \sum_{|N|=0}^{\infty} \frac{\psi_N(\boldsymbol{r})\psi_N(\boldsymbol{r}_0)}{k^2 - k_N^2 - jk_N/(\tau_N c)} \tag{2.13}$$

where τ_N is the time constant of mode N which depend on the wall admittance β. How to obtain τ_N will be described in Section 2.2.2. However, in practise one will use a finite summation of the modes in (2.13) to get an approximation of the model. Depending on the frequency range of interest, some of the modes will be negligible and thus the finite summation is sufficiently good approximation. An example of the Green's function in (2.13) is depicted in Figure 2.1.

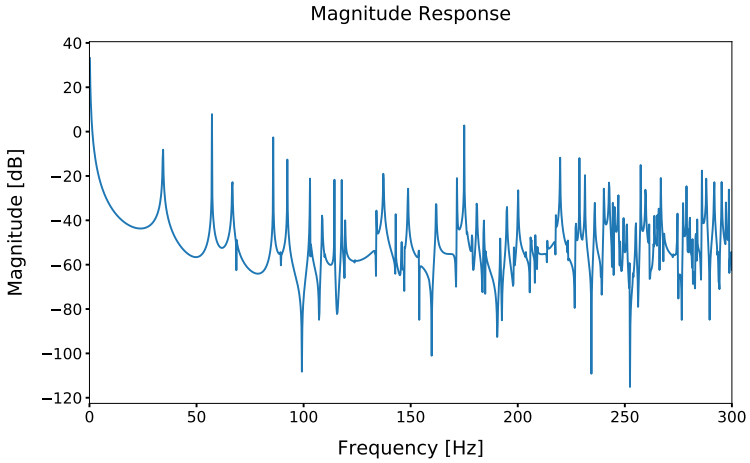


Figure 2.1: The magnitude response in dB of the Green's function (2.13). The room has the size $5 \times 3 \times 2\,m^3$, admittance $\beta = 2.05 \cdot 10^{-6} + 0.5j$, the source is placed at $\boldsymbol{r}_0 = [3.74,\ 2.23,\ 0.55]^\intercal$ and the microphone at $\boldsymbol{r} = [0.95,\ 1.79,\ 0.86]^\intercal$. The frequency axis is given by $f = \omega/2\pi$.

Now that the Green's function for the BVP in (2.10) have been presented, we have a model for the RTF of a room. In the following, the modes ψ_N and the corresponding eigenvalues k_N will be examined further, when we present some results from modal theory.

2.2 Modes

When a sound signal is emitted in a room, the signal excites the modes of the room. Especially, if the emitted sound signal only contains a single frequency close to a so called *resonance frequency*, the corresponding mode will dominate the sound field of the room. In this section, some of the theoretical aspects of modes will be given. Also, we will describe how modes affects an emitted sound signal and why this poses a problem in sound reproduction. The following is based on a combination of [13, Sec. 8.2] and [19, Sec. 3.2].

As already mentioned in the previous section, a eigenpair (k_N^2, ψ_N) consists of the mode ψ_N and corresponding wave number k_N. The resonance frequencies of a rectangular room can be computed by combining the equation for the wave numbers k_N in (2.12), and the relation between the wave number and frequency, $kc = 2\pi f$. Hence, we get the following:

$$f_N = \frac{k_N c}{2\pi} = \frac{c}{2\pi}\sqrt{\left(\frac{n_x\pi}{l_x}\right)^2 + \left(\frac{n_y\pi}{l_y}\right)^2 + \left(\frac{n_z\pi}{l_z}\right)^2} \tag{2.14a}$$

$$= \frac{c}{2}\sqrt{\left(\frac{n_x}{l_x}\right)^2 + \left(\frac{n_y}{l_y}\right)^2 + \left(\frac{n_z}{l_z}\right)^2} \tag{2.14b}$$

When a sound signal is emitted in the room and that carries one of the resonance frequencies of the room, we can conclude from (2.12) that the activation of the mode goes to infinity, i.e. for a fixed N and any $\boldsymbol{r}_0 \in \mathcal{B}$ with $\psi_N(\boldsymbol{r}_0) \neq 0$:

$$\frac{\psi_N(\boldsymbol{r}_0)}{k^2 - k_N^2} \longrightarrow \infty \quad \text{as} \quad k \longrightarrow k_N$$

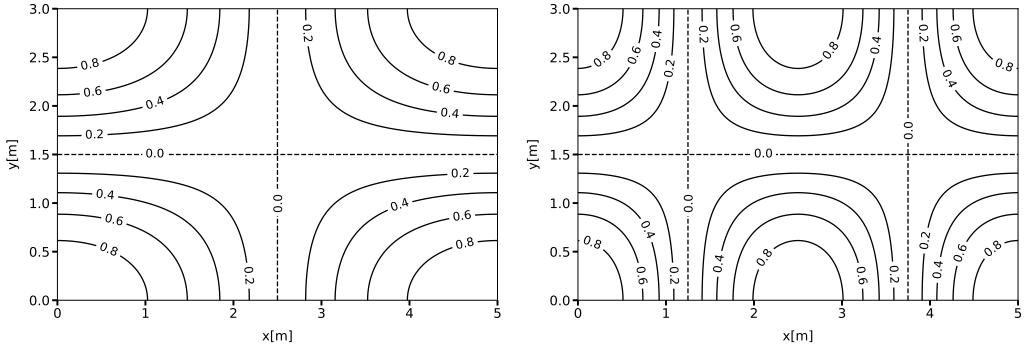
However, if $\boldsymbol{r}_0$ is chosen such that $\psi_N(\boldsymbol{r}_0) = 0$ the activation will be zero. When a room with non-rigid walls is considered instead, the above limit would be finite and complex. For a room with lightly damped walls, i.e. $\mathrm{Re}\{\beta\} \approx 0$, the magnitude of the activation will be relatively large [13, Sec. 8.4]. In Figure 2.1, some of the resonance frequencies of the room can be found by locating the positive peaks in the magnitude response. Depending on your position $\boldsymbol{r}$ or the source position $\boldsymbol{r}_0$ in the room, certain frequencies will be attenuated due to the behaviour of the modes, e.g. $\psi_N(\boldsymbol{r}) = 0$. Hence, the sound pressure of the modes will also be zero at these positions and will not be audible. From the definition of the modes ψ_N in (2.12), these positions are odd integers of the half wavelengths, which means they are in the set:

$$\boldsymbol{r} \in \left\{ \left(\frac{(2i-1)l_x}{2n_x}, y, z \right) \in \mathcal{B} \middle| \; i \in \mathbb{N} \right\} \cup \left\{ \left(x, \frac{(2i-1)l_y}{2n_y}, z \right) \in \mathcal{B} \middle| \; i \in \mathbb{N} \right\} \cup \left\{ \left(x, y, \frac{(2i-1)l_z}{2n_z} \right) \in \mathcal{B} \middle| \; i \in \mathbb{N} \right\} \tag{2.15}$$

The above set is an intersection of three sets of equidistant planes where the sound pressure will be zero. These planes are called *nodal planes* and a set as in (2.15) is often referred to as a *nodal set*. Note that the above nodal set only applies to rectangular rooms and is more complicated to obtain for arbitrary shaped rooms. This is because the nodal planes will no longer be planes but nodal surfaces [13, Sec. 8.2.1] [19, Sec. 3.2]. We will not be going further into the modal theory of arbitrary room shapes.

One distinguishes between different types of modes, depending on their *modal number* $N = (n_x, n_y, n_z)$. The trivial mode, also called the *cavity mode*, is the mode where all indices in N are zero [13, Sec. 8.2.1]. For this mode, the sound pressure is independent of the position in the given room (See Eq. (2.12) for the definition of ψ_N) and the air acts like a spring [13, Sec. 8.2.1]. The modes where two of the indices in N are zero, e.g. $(1, 0, 0)$, is referred to as *axial modes*. These modes corresponds to sound propagation in only one direction (See Eq. (2.12) for the definition of ψ_N). When only a single index in N is zero, e.g. $(1, 1, 0)$, the modes are called *tangential modes*. And finally, when all indices in the modal number are non-zero, the modes are know as *oblique modes*. [13, Sec. 8.2.1] [19, Sec. 3.2]

In Figure 2.2, the equal pressure contours for two different tangential modes have been depicted. From the figure, the nodal planes of the modes are shown as the zero-contour lines. It is not only the nodal planes that are important but also the locations where the sound pressure of the modes are at maximum.



(a) Modal number $(1, 1, 0)$. **(b)** Modal number $(2, 1, 0)$.

Figure 2.2: Equal pressure contours for two different modes in a rectangular room of size $5 \times 3 \times 2\, m^3$ (**a**) with the modal number $(1, 1, 0)$ and (**b**) with the modal number $(2, 1, 0)$. The contours are shown in the xy-plane, i.e. $z = 0$.

It can be seen from Figure 2.2, there are several positions where the sound pressure of a mode is high and therefore can be heard very clearly, e.g. the corners of the room in Figure 2.2a. Since our model of the sound field (2.13) do not just contain a few modes, no matter where you stand in a room at least one mode will be audible in that position. In terms of sound reproduction, this causes a problem, since frequencies close to the resonances will be amplified while others will be attenuated. This introduces undesired artefacts in the sound reproduction and will ruin the listener experience. In order to determine the maximum number of modes necessary for the approximation of the model, we will modal density in the following.

2.2.1 Modal Density

The modal density[1] describes the number of modes per hertz. In order to derive this density, one must have a function for counting the number of modes below a given frequency f. By restating (2.14) as an inequality, one obtains:

$$\left(\frac{cn_x}{2l_x}\right)^2 + \left(\frac{cn_y}{2l_y}\right)^2 + \left(\frac{cn_z}{2l_z}\right)^2 \leq f^2 \tag{2.16}$$

which corresponds to a closed ball with center at $\mathbf{0}$ and radius f, i.e. $\bar{B}(\mathbf{0}, f)$. Hence, the number of modes below f can be stated as the function:

$$N(f) = \left|\left\{(n_x, n_y, n_z) \in \mathbb{N}^3 \;\middle|\; \left(\frac{cn_x}{2l_x}, \frac{cn_y}{2l_y}, \frac{cn_z}{2l_z}\right) \in \bar{B}(\mathbf{0}, f)\right\}\right| \tag{2.17}$$

The function above counts all the modes below f but we can derive another expression of (2.17) by counting axial, tangential, and oblique modes separately. From (2.16) it follows that the number of oblique modes can be obtained from counting the number of rectangular boxes with dimension $(c/2l_x, c/2l_y, c/2l_z)$ within $\bar{B}(\mathbf{0}, f)$. However due to symmetry, it is only necessary to count the number of rectangular boxes within an spherical octant of radius f (See Figure 2.3) [13, Sec. 8.2.2]. Hence, the number of oblique modes below f is given by:

$$N_{\text{obl}}(f) = \frac{1}{8}\frac{\left|\bar{B}(\mathbf{0}, f)\right|}{\frac{c}{2l_x}\frac{c}{2l_y}\frac{c}{2l_z}} \tag{2.18a}$$

$$= \frac{1}{8}\frac{8V}{c^3}\frac{4\pi}{3}f^3 \tag{2.18b}$$

$$= \frac{4\pi V}{3c^3}f^3 \tag{2.18c}$$

[1] This is not a density in the sense of probability, since it does not integrate to one.

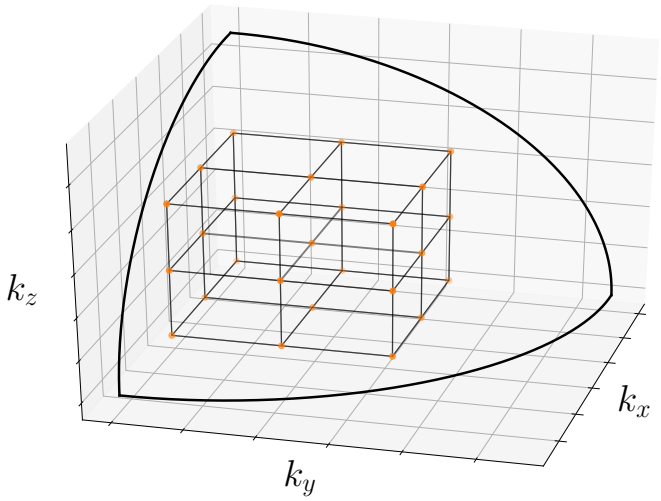


Figure 2.3: Illustration of counting the number of oblique modes below f. The wavenumbers are given on the axes, e.g. $k_x = \frac{\pi n_x}{l_x}$.

where V is the volume of the room. The number of axial and tangential modes can be computed in a similar manner. The number of axial modes corresponds to the number of points on each axis in the interval $[0, f]$ and is given as:

$$N_{\text{axs}}(f) = \frac{2l_x}{c}f + \frac{2l_y}{c}f + \frac{2l_z}{c}f = \frac{L}{2c}f \tag{2.19}$$

where $L = 4\left(l_x + l_y + l_z\right)$ [13, Sec. 8.2.2]. The number of tangential modes corresponds to counting the number of squares within a quarter of a circle with radius f for each plane:

$$N_{\text{tan}}(f) = \frac{1}{4}\frac{\pi f^2}{\frac{c}{2l_x}\frac{c}{2l_y}} + \frac{1}{4}\frac{\pi f^2}{\frac{c}{2l_y}\frac{c}{2l_z}} + \frac{1}{4}\frac{\pi f^2}{\frac{c}{2l_x}\frac{c}{2l_z}} \tag{2.20a}$$

$$= \frac{\pi l_x l_y}{c^2}f^2 + \frac{\pi l_y l_z}{c^2}f^2 + \frac{\pi l_x l_z}{c^2}f^2 \tag{2.20b}$$

$$= \frac{\pi S}{2c^2}f^2 \tag{2.20c}$$

where $S = 2\left(l_x l_y + l_x l_z + l_y l_z\right)$ is the surfaces area of all the walls. The number of axial, tangential, and oblique modes below f is proportional to f, f^2, and f^3 respectively [13, Sec. 8.2.2]. Thus, the number of oblique modes will be dominating the sound field, except for lower frequencies, and (2.17) can be approximated by [13, Sec. 8.2.2]:

$$N(f) \simeq \frac{4\pi V}{3c^3}f^3 \tag{2.21}$$

The modal density is given by the derivative of $N(f)$ and can then be approximated by the derivative of (2.21):

$$n(f) = \frac{dN(f)}{df} \simeq \frac{4\pi V}{c^3} f^2 \tag{2.22}$$

It can be shown that the above density is asymptotically valid in any room regardless of its shape [13, Sec. 8.2.2]. For the rectangular room, a more accurate expression can be derived by including (2.19) and (2.20). Since an octant of a ball was used to compute (2.18), the points in the planes will be counted as half points and those on the axes as quarters [19, Sec. 3.2]. Adding these corrections yield:

$$N(f) = \frac{4\pi V}{3c^3} f^3 + \frac{1}{2}\frac{\pi S}{2c^2} f^2 + \frac{1}{4}\frac{L}{2c} f \tag{2.23}$$

Therefore, the modal density for the rectangular room is given by:

$$n(f) = \frac{4\pi V}{c^3} f^2 + \frac{\pi S}{2c^2} f + \frac{L}{8c} \tag{2.24}$$

From (2.21) and (2.22), it is evident that as the number of modes increases the modes will become dense in frequency. This is one of the reasons why our proposed model discussed above in Section 2.1.2 is less useful for higher frequencies. In Section 2.1.2, it was discussed that (2.13) could be approximated by a finite sum since some of the modes would be negligible. The number of necessary summands can be estimated by $N(f_{max})$, where f_{max} is the highest frequency of interest. So for a high frequency range a large number of complex terms are summed and our model will become sensitive to small errors in each term, e.g. change in dimensions of the room or change in propagation speed [13, Sec. 8.3]. However, this is only of concern when the Green's function is used for modelling sound fields in real rooms. In this project, the model will be used for simulating RTFs and generate data sets, thus we control the room dimensions, the propagation speed, etc.

2.2.2 Reverberation Time And Damping Constant

As mentioned previously in Section 2.1.2, when the sound field in a room with lightly damped walls are considered, the terms of the Green's function consist of a complex term as part of the denominator. This complex term depends on a time constant of the mode, τ_N. In this section, the time constant will be related to the *reverberation time* of a room. Also, a method for computing this time constant will be stated for the rectangular room.

First, let us consider a single series term of the Green's function (2.13). Taking the absolute value yields:

$$\left| -\frac{1}{V} \frac{\psi_M(\boldsymbol{r})\psi_M(\boldsymbol{r}_0)}{k^2 - k_M^2 - jk_M/(\tau_M c)} \right| = \frac{1}{V} \frac{|\psi_M(\boldsymbol{r})\psi_M(\boldsymbol{r}_0)|}{\left[\left(k^2 - k_M^2\right)^2 + k_M^2/(\tau_M^2 c^2) \right]^{1/2}} \tag{2.25}$$

for some given $M \in \mathbb{N}_0^3$. From the above magnitude response, it can be shown that the 3dB-bandwidth for mode M is given by [13, Sec. 8.3] [19, Sec. 3.4]:

$$(\Delta f)_M = \frac{1}{2\pi\tau_M} = \frac{3\log(10)}{\pi T_{rev}} \tag{2.26}$$

The 3dB-bandwidth of a mode describes the width of the frequency band where the magnitude response is greater than half its maximum value [13, Sec. 8.3] [19, Sec. 3.4]. The mode is said to be excited if the frequency band of a narrow band signal falls into this 3dB-band [13, Sec. 8.3]. The constant T_{rev} is the reverberation time of the room. Formally, it is defined as the time from a sound source have been turned off to the sound pressure in the room have decreased 60dB. In general, the reverberation time is related to the time constant τ_M by [13, Sec. 8.3] [19, Sec. 3.4]:

$$T_{rev} = 6\log(10)\bar{\tau} \tag{2.27}$$

where $\bar{\tau}$ denotes the average over time constants [19, Sec. 3.4]. The reverberation time for a given room can be measured, and then the average time constant, $\bar{\tau}$, can be obtained from (2.27), which can be used as an estimate for τ_M [13, Sec. 8.3]. However, if the dimensions of the room are known and assuming that all surfaces have the same wall admittance, β, then the time constant τ_M can be estimated as follows [13, Sec. 8.4.1]:

$$\tau_M \simeq \begin{cases} \dfrac{V}{2cS\,\mathrm{Re}\{\beta\}} & \text{for oblique modes} \\ \dfrac{3V}{5cS\,\mathrm{Re}\{\beta\}} & \text{for tangential modes} \\ \dfrac{3V}{4cS\,\mathrm{Re}\{\beta\}} & \text{for axial modes} \end{cases} \tag{2.28}$$

where V is the volume of the room and S is the surface area of the walls. The estimate for tangential and axial modes are obtained by further assuming that the dimensions are almost equal, i.e. $l_x \simeq l_y \simeq l_z$ [13, Sec. 8.4.1].
A similar estimate of τ_M can be obtained using a statistical approach. The estimate will only be stated here but the reader are referred to Jacobsen et al. [13, Sec. 8.4.2] for the details:

$$\tau = \frac{4V}{cA} \tag{2.29}$$

where the subscript M have been dropped, since the estimate does not depend on the mode number, and A is the total absorption area. Under the assumption of uniform absorption, we have $A = S\alpha$ where α is the absorption coefficient of the walls. Comparing (2.29) and the estimate of the time constant for oblique modes in (2.28), one observes that these are equal for:

$$\alpha = 8\,\mathrm{Re}\{\beta\}$$

By definition of the absorption coefficient $0 \leq \alpha \leq 1$ [13, Sec. 8.3.1]. Thus, we can put constraints on the on the wall admittance, which are consistent with the assumption $|\beta| \ll 1$:

$$0 \leq \mathrm{Re}\{\beta\} \leq \frac{1}{8} \tag{2.30}$$

Note that the upper bound in (2.30) violates the assumption of lightly damped walls. Mathematically there is no problem with (2.30), but the modal decomposition are not guaranteed to reflect the true physical behaviour of the RTF. We will return to this issue in the next chapter when generating data sets.

In Section 2.2.1, it was argued that the number of oblique modes will be dominating the sound field. Thus, we have decided to use the oblique estimate of the time constant with the constraints given (2.30), when generating RTFs for our data sets.

The Schroeder Frequency

We now return to the short discussion of Section 2.2.1. The Green's function turned out to be sensitive to errors at high frequencies when modelling real rooms. The Schroeder frequency is the upper limit for which one will use the Green's function in practise, and statistical models should be considered for higher frequencies. Before stating the Schroeder frequency, the *modal overlap* is defined [13, Sec. 8.3]:

$$M_O(f) = n(f)\Delta f \tag{2.31a}$$

$$\simeq \frac{4\pi V}{c^3} f^2 \frac{3\log(10)}{\pi T_{rev}} \tag{2.31b}$$

$$= \frac{12\log(10)V}{c^3 T_{rev}} f^2 \tag{2.31c}$$

The modal overlap is the average number of modes excited by a narrow band signal assuming that each mode is excited only if the frequency band of the excitation signal is within the 3dB-bandwidth Δf centred at the corresponding resonance frequency [13, Sec. 8.3]. If the modal overlap is smaller than or equal to one, the modes are separable, i.e. only one mode are excited at the time [19, Sec. 3.4]. On the other hand, if the modal overlap is greater than one, multiple modes will be excited at the same time [19, Sec. 3.4]. According to Schroeder, a modal overlap greater than three is sufficient to justify a statistical model instead of the Green's function [13, Sec. 8.3]. Setting (2.31) equal to three and isolating f, we obtain the Schroeder frequency:

$$f_{Sch} = \sqrt{\frac{c^3 T_{rev}}{4\log(10)V}} \tag{2.32}$$

When generating RTFs for the data sets, we should be aware that the Schroeder frequency for the simulated rooms are above 300Hz. If this is not the case, the relatively high modal overlap would introduce errors for the simulated RTFs in the

frequency range of interest. Considering a real component of the wall admittances in the higher end of the interval given in (2.30), i.e. $\mathrm{Re}\{\beta\} \approx 1/8$, would result in a relatively low Schroeder frequency. For example, a room with a volume of $30\,m^3$ and $\mathrm{Re}\{\beta\} = 1/200$ results in a Schroeder frequency of approximately 534Hz, while for the same room volume and $\mathrm{Re}\{\beta\} = 1/8$ the Schroeder frequency become approximately 107Hz. However, we deem that this is not necessarily a big problem since the CNN models might learn to ignore the errors, but there is no guarantee that this is the case.

In this chapter, a short introduction on the theory of sound fields in enclosures where given. The Green's function for rectangular rooms with non-rigid walls was stated, and in the following, we will present how this is used for generating RTFs.

Chapter 3

Sound Field Simulation and Generation of Data Sets

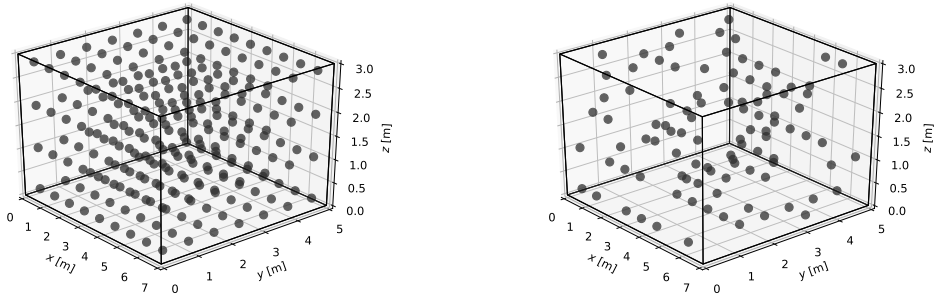
In the previous chapter, the theory of room acoustics and sound fields in enclosures were presented. This theory will be used to make scripts for simulating and generating realistic Room Transfer Functions (RTFs) for training and testing the Convolutional Neural Network models (See Section 5.1). In this chapter, a short description of how the RTFs are simulated and generated using our implemented module `SoFIE` (See Appendix A) is given. Various test have been conducted to verify the consistency of the RTFs with the theory presented in Chapter 2, and the result of these tests will be given here as well. We end this chapter with a description of how the various data sets have been generated and sorted into different *Data set Collections (DSCs)*.

3.1 Module for Simulating Room Transfer Functions

Based on the theory introduced in Chapter 2, a Python module called `SoFIE` (**So**und **F**ields **I**n **E**nclosures) have been developed for simulating RTFs inside a given room. In order to generate RTFs with this module, first the characteristics of a room must be defined before simulating the sound field. The necessary room characteristics are the room dimensions and the wall admittance β of the surfaces (See Section 2.1.2), which is assumed to be the same for all surfaces. The Python class `Room` in `SoFIE` contains the necessary functions for creating different types of spatial grids, which are used for making the microphone positions in the room (See Figure 3.1), as well as a function for generating the RTFs.

In order to use the functions for creating the microphone grids, one must specify the grid parameters first, i.e. the grid size in meters and number of grid points along each axis. The grid size refers to the dimension hereof, i.e. the length, width and height of the grid. For instance, if the grid size equals the room dimensions, the grid points will be placed in the interior of the room as well as the walls. The `Room` class

can create a cubic lattice grid for microphones. A set of grid points have the size of $D \times M$, where M is the total number of grid points and $D = 3$ is the dimension of the spatial field. The position of the grid points are defined in the internal coordinate system of the room. The origin of the coordinate system is placed in a corner of the room. From the origin the x-axis is defined along the length, the y-axis along the width, and the z-axis along the height of the room. The function for making a cubic lattice are used in generating the microphone grids for the DSCs. Hence in order to make a random placed microphone configuration, a random subset will be sampled from a dense microphone grid. An illustration of a dense grid and a random subset hereof are shown in Figure 3.1.



(a) Dense grid.

(b) Random subset of the dense grid.

Figure 3.1: An illustration of (**a**) a dense grid in a given room of $7 \times 5 \times 3\, m^3$ and (**b**) a random subset of the dense grid. The dense grid is made by using the cubic lattice grid function in the module `SoFIE`.

Besides the positions of the microphones and sources, a sampled frequency-axis f_a is necessary for generating the RTFs. This can be obtained from the Discrete Fourier Transform given a sampling-rate, f_s, and an excitation time, t_s:

$$f_a := \begin{cases} \left\{ \frac{i}{t_s} \right\}_{i=0}^{n/2} & \text{for } n = \lfloor t_s f_s \rfloor \text{ even.} \\ \left\{ \frac{i}{t_s} \right\}_{i=0}^{(n-1)/2} & \text{for } n = \lfloor t_s f_s \rfloor \text{ odd.} \end{cases} \tag{3.1}$$

The frequency-axis is computed using `numpy`'s `fft` module for generating the DSCs. The frequency resolution depends on the sampling-rate and the sampling time interval. The frequency axis is limited to the lower frequency range, i.e. 15 - 300 Hz (See the Problem Statement p. 5). Thus, the sampling-rate is fixed to $f_s = 600$ Hz to satisfy Nyquist-Shannon's sampling theorem. The frequency resolutions then only depend on the excitation time, t_s (See Section 1.3.3).

When all the necessary parameters of the room, and the positions of microphones and sources have been given, the RTFs are computed for all the given pairs of microphone and source positions using the `RTF` function in `SoFIE`. The function is an implementation of the Green's function for rooms with non-rigid walls (See Section 2.1.2) using a finite number modes in the summation. The function uses an estimate of the time constant τ_N given by (2.28), based on the room dimensions and the wall admittances. Before this module are used for simulating the RTFs and generating the DSCs, the programmed simulation module, `SoFIE`, is verified through tests.

3.2 Verification of Simulated Data

In the following, the reliability of the simulation module `SoFIE` is examined. In order to ensure the simulated RTFs created by the module are realistic and follow the presented theory in Chapter 2 (See also [13, Ch. 8]). For these tests, some the variables for generating the RTFs have been fixed such as the room dimensions and the real component wall admittance, $\mathrm{Re}\{\beta\}$. The fixed variables and their values are given in Table 3.1.

Variables	Room size $[m^3]$	$\mathrm{Re}\{\beta\}$	Sampling Rate $[Hz]$	Excitation time $[s]$
Values	$5 \times 3 \times 2$	$2.05 \cdot 10^{-6}$	600	5

Table 3.1: The fixed variables used for the tests of the SoFIE-module.

3.2.1 Symmetric Property of the Room Transfer Function

The first test was to examine the symmetric property of the RTF. The symmetric property refers to that the RTF for a given source- and microphone position is equal to the RTF of interchanged source- and microphone position (See Section 2.1.2). Hence, we first randomly place a single source and a single microphone inside the room $\boldsymbol{r}_s$ and $\boldsymbol{r}_m$ respectively. The placement of the microphone and the source are drawn from a three-dimensional uniform distribution, $\mathcal{U}(\mathcal{B})$, where $\mathcal{B} := (0,5) \times (0,3) \times (0,2)$ is the room including the surfaces. An RTF is then generated for the microphone and source position, and an other for the interchanged positions. The two RTFs are stored as arrays and to avoid issues due to the numerical precision, we examine the difference entrywise:

$$\left|\hat{G}_i(\boldsymbol{r}_m, \boldsymbol{r}_s) - \hat{G}_i(\boldsymbol{r}_s, \boldsymbol{r}_m)\right| < \epsilon \qquad \forall i \in \mathcal{I} \tag{3.2}$$

where $\mathcal{I} = \{0, 1, \cdots, |f_a|\}$ is the index-set of the arrays, f_a is the frequency axis (See Equation (3.1)), and $\hat{G}(\boldsymbol{r}_m, \boldsymbol{r}_s)$ and $\hat{G}(\boldsymbol{r}_s, \boldsymbol{r}_m)$ are the two generated RTFs. We found that the maximum absolute difference over the entries was on the magnitude of 10^{-15}. Due to the decimal precision of the employed computers, the difference in (3.2) is numerical zero for every index i. Thus, the RTFs are equal and the symmetric property holds for our simulated data.

3.2.2 Visual Spoting of the Resonance Frequencies

The succeeding test was to examine the property of the RTF that the locations of the resonance frequencies are visible from the magnitude response of the RTF given a single source and microphone placed in the opposite corners of a given room. For this test, the source is placed in the corner $\boldsymbol{r}_s = \mathbf{0}$ and the microphone in the opposite corner $\boldsymbol{r}_m = [5\ 3\ 2]^\intercal$. The location of the resonance frequencies can then be compared with the theoretical values of the eigenfrequencies, which can be calculated from the mode numbers as stated in Section 2.2. However, multiple mode numbers could result in the same eigenfrequency and these might cancel each other in the RTF. In order to illustrate this, an example from the test is given in the following.

From (2.13), we have that the product $\psi_N(\boldsymbol{r_s})\psi_N(\boldsymbol{r_m})$ is included in the summation for computing the RTF, where $\psi_N(\boldsymbol{r_s})$ is the mode evaluated at the source and $\psi_N(\boldsymbol{r_m})$ is the mode evaluated at the microphone position. The modes themselves are given by a product of three cosines (See Section 2.1.2). Hence, it easily follows that $\psi_N(\boldsymbol{r_s}) = \sqrt{\epsilon_{n_x}\epsilon_{n_y}\epsilon_{n_z}}$ (See Section 2.1.2). For the microphone we get:

$$\psi_N(\boldsymbol{r_m}) = \sqrt{\epsilon_{n_x}\epsilon_{n_y}\epsilon_{n_z}}\cos\left(\frac{n_x\pi}{5}\cdot 5\right)\cos\left(\frac{n_y\pi}{3}\cdot 3\right)\cos\left(\frac{n_z\pi}{2}\cdot 2\right) \tag{3.3a}$$

$$= \sqrt{\epsilon_{n_x}\epsilon_{n_y}\epsilon_{n_z}}\cos(n_x\pi)\cos(n_y\pi)\cos(n_z\pi) \tag{3.3b}$$

$$= \sqrt{\epsilon_{n_x}\epsilon_{n_y}\epsilon_{n_z}}(-1)^{n_x+n_y+n_z} \tag{3.3c}$$

Thus, the sign of the summand in (2.13) are determined by the mode number. From this test, we found that multiple mode numbers gave the same eigenfrequencies, e.g. the mode numbers $(1, 3, 0)$ and $(1, 0, 2)$ both give the eigenfrequency 175.01 Hz. If we compute the signs using (3.3c), we get 1 and -1 respectively. Hence, these cancel each other and the eigenfrequency 175.01 Hz will not be present as a resonance frequency in the RTF.

Given that multiple mode numbers can results in the same eigenfrequncy and these might cancel each other out. The eigenfrequencies that cancel out should be removed from the set of resonance frequencies, we expect to visual spot in the RTF. We will now examine the magnitude response of the RTF generated for the given source and microphone position. The magnitude response can be seen in Figure 3.2.

As seen from Figure 3.2, the magnitude response of the RTF peaks at certain frequencies, which correspond to the resonance frequencies. Also, the number of peaks increases with the frequency, which is consistent with the theory (See Section 2.2). In order to examine the magnitude response further, the frequency-axis have been split into four sub-axis as seen in Figure 3.3. In the sub figures, vertical lines have been added to illustrate where the theoretical resonance frequencies are.

From Figure 3.3, it can be observe that the vertical lines and the peaks of the magnitude response aligns almost perfectly. The few places where they do not seem

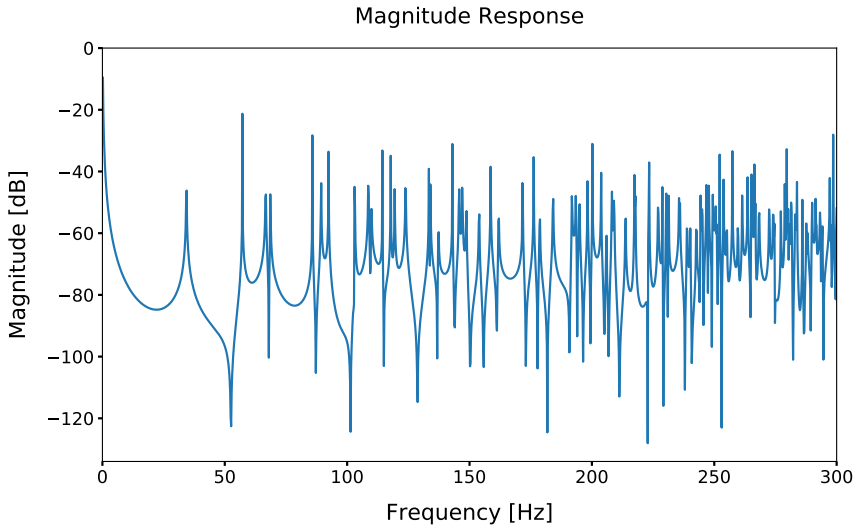


Figure 3.2: The magnitude response of the RTF in dB. The source and microphone are placed in opposite corners of the room, i.e. $\boldsymbol{r}_s = \mathbf{0}$ and $\boldsymbol{r}_m = [5\ 3\ 2]^\intercal$ respectively. The room have the size $5 \times 3 \times 2\,m^3$ and the real component of the wall admittance, $2.05 \cdot 10^{-6}$.

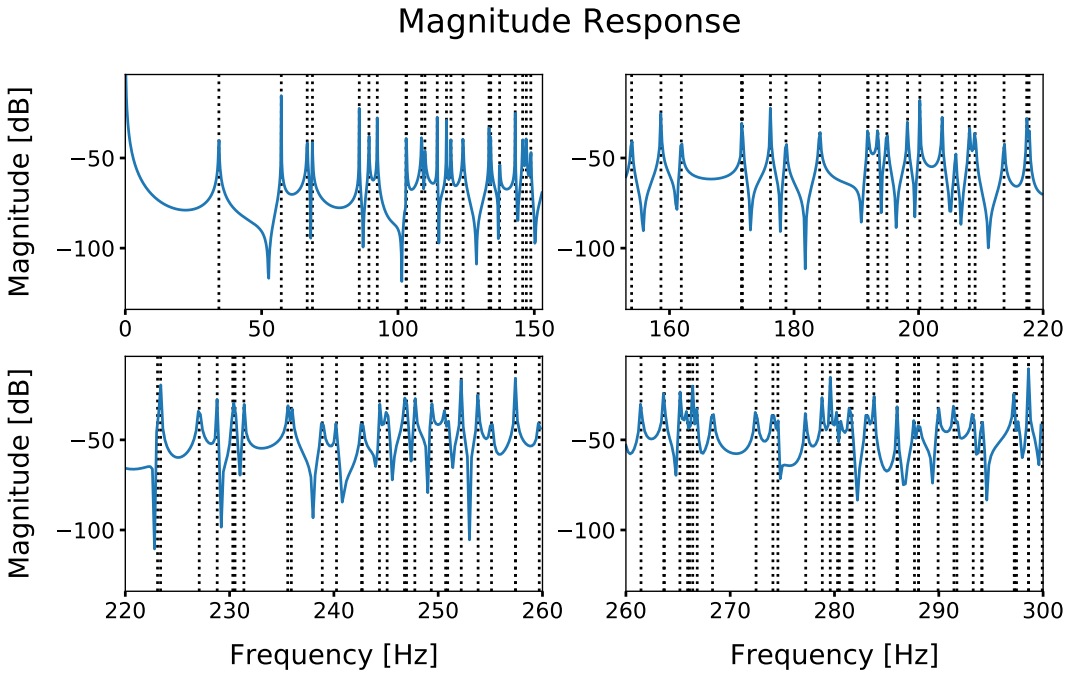


Figure 3.3: The magnitude response from Figure 3.2, where the frequency axis have been divide into four subsets. The dashed, vertical lines represent the theoretical calculated resonance frequencies of the room.

to align, are due to the resolution and the fact that the resonance frequencies tends

to get dense for higher frequencies. Hence, we conclude that the simulated data satisfies the desired property regarding the resonance frequencies.

3.2.3 Pressure Contours

The final test is to examine the pressure contours in a given plane when the room is excited by a resonance frequency. From the theory of sound fields, we know how these contours should look like for a given resonance frequency and mode number. For a given mode number (n_x, n_y, n_z), the integers n_x, n_y and n_z will determine how many positions on a given axis the pressure will be zero. E.g. $n_x = 3$ means that there are three positions on the x-axis where the pressure will be zero. In this test, the source will be placed in the corner, $\boldsymbol{r}_s = \mathbf{0}$, as before in Section 3.2.2. In order to create the contours, a dense grid of microphones are generated in the xy-, xz- or yz-plane. The planes are given by fixating one coordinate axis to zero, e.g. for the xy-plane z is zero.

In Figure 3.4, the pressure contours for the resonance frequency 89.85 Hz are depicted in the xy-plane. The white dashed lines in the figure indicate where the sound pressure should be equal to zero theoretically.

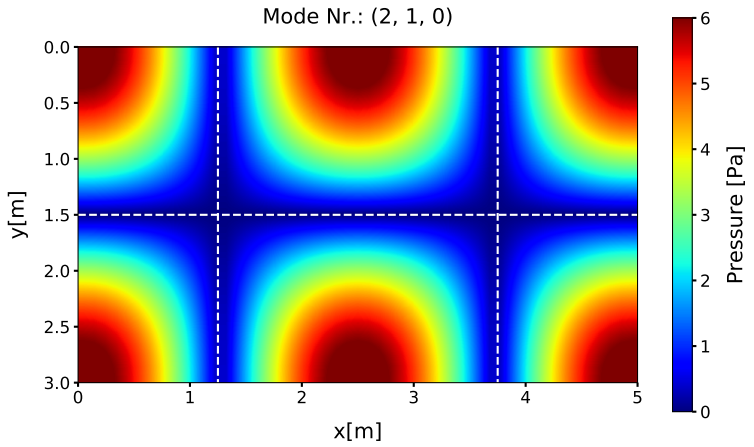
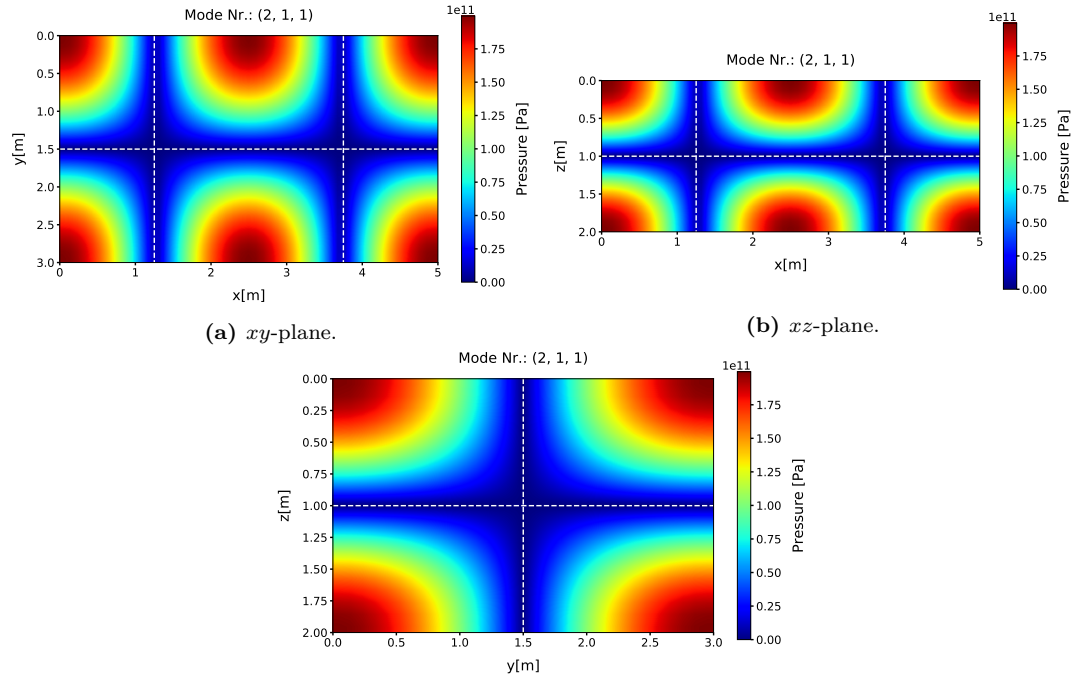


Figure 3.4: Pressure contours in the xy-plane. The excitation frequency was the resonance frequency 89.35 Hz corresponding to the mode number $(2, 1, 0)$. The source is located in the corner, $\mathbf{0}$, the room have the size $5 \times 3 \times 2\,m^3$ and the real component of the wall admittance, $2.05 \cdot 10^{-6}$.

As seen from Figure 3.4, the white lines align with the contours exactly where the sound pressure is zero. However, this is only for a two dimensional mode number or a tangential mode number (See [13, Sec. 8.2.1]). Next, the contours for a three dimensional mode number, also called a oblique mode number, is examined (See Section 2.2). In Figure 3.5, the pressure contours for the resonance frequency 123.88 Hz are depicted in all there planes.



(a) xy-plane.

(b) xz-plane.

(c) yz-plane.

Figure 3.5: Pressure contours in all three planes. The excitation frequency was the resonance frequency 123.88 Hz corresponding to the mode number $(2, 1, 1)$. The source is located in the corner, $\mathbf{0}$, the room have the size $5 \times 3 \times 2\,m^3$ and the real component of the wall admittance, $2.05 \cdot 10^{-6}$.

Again as before, the white lines align with the contours where the sound pressure is zero. Hence, we can conclude that the module also satisfy this test and the module can be used for generating reliable data.

3.3 Generating Data Set Collections

To get an overview of the different data sets we have generated for this project, we will in the following give an introduction to the data generation process.

The overall structure of the data sets follow a similar pattern. We sort the data sets into different collections by the combination of data generation set-up and microphone configuration, such that the only difference between data sets in a single collection is the number of microphones.

The chosen data generation set-ups are: varying room dimensions and fixed wall admittance, varying wall admittances and fixed room dimensions, and varying room dimensions and wall admittances. For each of these set-ups, we construct different microphone configurations. A total number of three different microphone configura-

tions have been chosen for generating the DSCs. The different microphone configurations will be described in the following. Hence, by training a CNN model with a single DSC, the impact of the number of microphones on the results can be studied, as well as the results for different DSCs can be compared.

3.3.1 Cubic Grid of Microphones Centred in the Room

The first microphone configuration is a cubic grid of microphones centred in the middle of the room, which will be referred to as a *centred grid*. The microphones in the centred grid are uniformly spaced and an example of such a grid is illustrated in Figure 3.6 for 27 microphones.

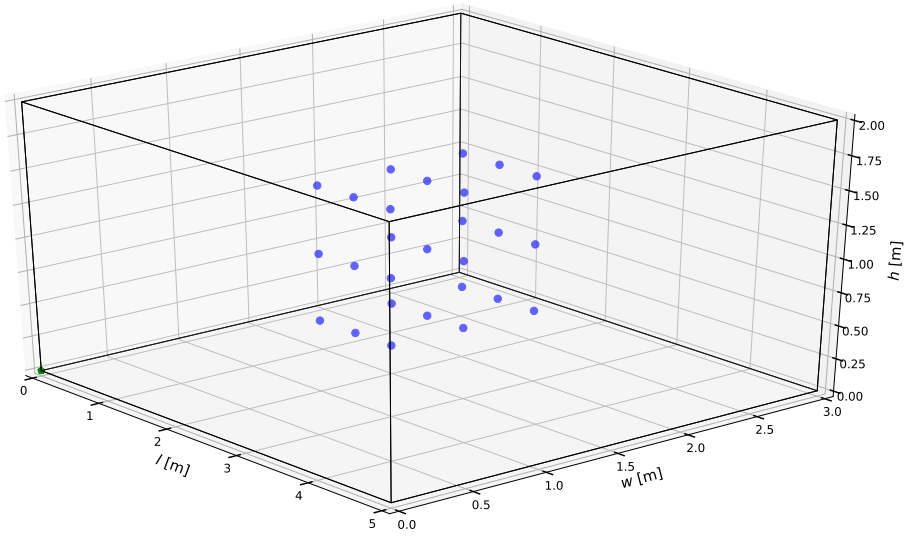


Figure 3.6: Illustration of $M = 27$ microphones (**blue dots**) placed in a cubic lattice grid, which is centred in the middle of the room and its called a centred grid. The source (**green pentagon**) is placed in the corner **0**.

The different number of microphones chosen for making the different data sets in each DSC are $M \in \{1, 8, 27, 64, 125\}$. The reason for these choices are that they can be easily uniformly spaced in a symmetrical cubic lattice grid. E.g. 8 microphones correspond to the corner points of a cube. For the special case where there is only a single microphone, the microphone is placed in the point $(1, 1, 1)$ for all rooms instead of the middle of the rooms. This is because valuable information regarding the odd-numbered modes will be lost, since their magnitude will be zero in the middle of the room (See Section 2.2).

We have chosen the size of the grid to be $1 \times 1 \times 1\, m^3$ independent of the number of microphones and room dimensions, such that our cubic lattice grid can be placed

even in the smallest room considered in the DSCs. The cubic microphone grids are made using the `SoFIE` module as mentioned in Section 3.1. The module generates the grid points in a specific order, which is independent on the size of the grid or the room itself. Thus, the CNN models might be able to exploit this pattern during the training, which could help guiding the model.

3.3.2 Cubic Grid of Microphones Centred in Different Positions

The second microphone configuration is a cubic grid of microphones centred in different positions. This configuration is a translation of the centred grid in Section 3.3.1 to different positions in the room and will be referred to as a *translated grid*. The microphone grid is translated to other positions by choosing one out of the positions from a second grid as its centre instead of the middle of the room as above. This second grid used for translating the microphone grid is a cubic lattice grid with the size $l-1.1 \times w-1.1 \times h-1.1\,m^3$. This size of the cubic lattice grid have been chosen, such that none of the microphones in the microphone grid will be placed in the walls or outside of the room. An illustration of a cubic lattice grid with 8 points and the microphone grid centre in one of them is shown in Figure 3.7.

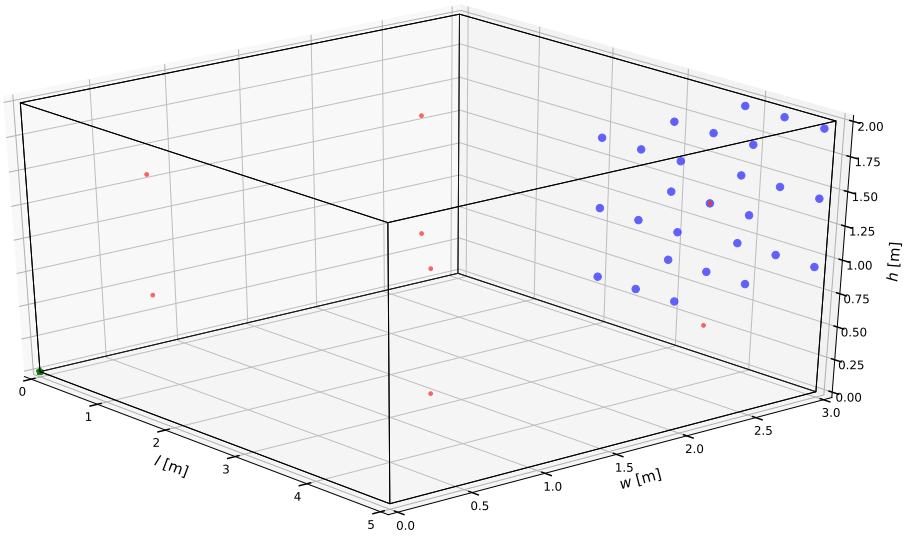


Figure 3.7: Illustration of an 8 point cubic lattice grid (**red points**), where one out of its points is used as centre for the microphone grid with $M = 27$ microphones (**blue dots**). The source (**green pentagon**) is placed in the corner **0**.

3.3.3 Randomly Placed Microphones

The last microphone configuration is randomly placed microphones and will be referred to as a *random subset*. The random placement of the M microphones is made

by sub sampling a dense centre grid, as defined in Subsection 3.3.1. The dense grid is a cubic lattice with the size $0.9l \times 0.9w \times 0.9h\ m^3$. To ensure randomness with $M = 125$ microphones, we have chosen the dense grid to consist of 512 positions. An illustration of 27 random placed microphones from a sub sampling of a dense grid with 125 points is shown in Figure 3.8.

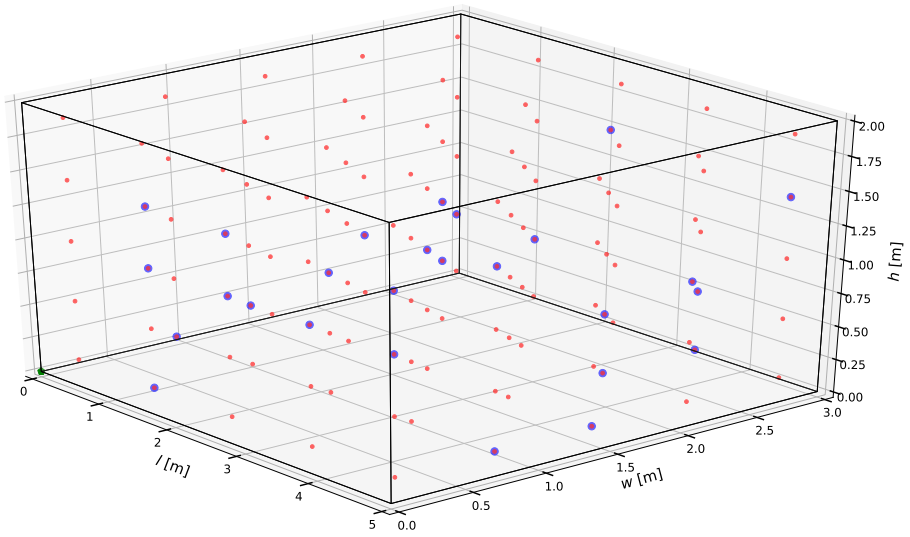


Figure 3.8: Illustration of $M = 27$ randomly placed microphones (**blue dots**) made by sub sampling a dense grid of 125 points (**red points**). The source (**green pentagon**) is placed in the corner **0**.

The three microphone configurations will be use for making different DSCs of each of the data generation set-ups. In the following, the fixed parameters and constraints on the room dimensions and the wall admittance will be given.

3.3.4 Fixed Parameters and Constraints

In the Problem Statement, the frequency range has been limited to $[15, 300]$ Hz (See p. 5) and the chosen sampling rate is 600 Hz as mentioned above in Section 3.1. An excitation time of 3 seconds have been chosen. Thereby, each recording from a microphone will consist of 855 frequency samples (See Section 1.3.3). Furthermore, the source is placed in the corner of the room corresponding to the origin, **0**, of the room coordinate system as described in Section 1.3.3. These parameters will be kept fixed for all the data sets.

From the Problem Statement, the room shapes are restricted to ideal rooms (See p. 5). The dimensions of the different rooms have been chosen such that they follow the EBU Tech 3276 [6] standard for sound control rooms regarding the volume not

exceeding $300\,\mathrm{m}^3$. Furthermore, the proportions of the room must satisfy the following limits of the length to height and the width to height ratio to ensure uniform distribution of the lower frequency eigenmodes [6]:

$$\frac{1.1w}{h} \leq \frac{l}{h} \leq \frac{4.5w}{h} - 4 \tag{3.4}$$

where l is the length, w is the width, and h is the height of the room. Furthermore, the inequalities $l < 3h$ and $w < 3h$ must apply. Moving terms around and we obtain the following:

$$1.1w \leq l \leq 4.5w - 4h \tag{3.5}$$

From (3.5), we obtain a lower bound for the length of the room. Thus, we got the following constraints $1.1w \leq l < 3h$. Given these boundaries for l, it follows that w has a smaller upper bound, i.e. $w < 3/1.1h$. By considering equality in the first inequality of (3.5), we get:

$$1.1w \leq 4.5w - 4h \tag{3.6}$$

Hence, we obtain a lower bound $w \geq 4/3.4h$ and thereby the constraints $4/3.4h \leq w < 3/1.1h$. We use these boundaries to create different rooms as shown in Algorithm 1.

Algorithm 1 Room Construction

$h_{min} \leftarrow$ Minimum height
$h_{max} \leftarrow$ Maximum height
$N_h \leftarrow$ Number of different heights
$N_w \leftarrow$ Number of different widths
$N_l \leftarrow$ Number of different lengths
$heights \leftarrow h_{min} + (h_{max} - h_{min})\frac{i}{N_h - 1}, \quad i = 0, 1, \ldots, N_h - 1$
for h in $heights$ **do**
    $widths \leftarrow \frac{4}{3.4}h + \left(\frac{3}{1.1} - \frac{4}{3.4}\right)\frac{ih}{N_w}, \quad i = 0, 1, \ldots, N_w - 1$
    for w in $widths$ **do**
        $lengths \leftarrow 3h + (1.1w - 3h)/N_l, \quad i = 0, 1, \ldots, N_l - 1$
        for l in $lengths$ **do**
            $room\ dimensions \leftarrow [l, w, h]$
return $room\ dimensions$

The total number of rooms is given as the product of N_l, N_w, and N_h, which are the number of different lengths, widths and heights respectively.

As there exist constraints for the dimensions of the rooms, we also have some limitation on the wall admittance given in Section 2.1.2, which states that the real component of the wall admittance β should be between $0 \leq \mathrm{Re}\{\beta\} \leq 1/8$. However,

as mentioned above in Section 2.2.2, choosing $\mathrm{Re}\{\beta\} \approx 1/8$ violates the assumption of lightly damped rooms (See Section 2.1.2), which the modal decomposition are based upon. In order to generate a data set with enough variety in the admittance value, we have chosen to use the following constraints despite the violations. The purpose of the CNN models is to classify the admittance values, so we deem that the violations of the theoretical assumptions are a minor problem. Thus, N_b real values of the wall admittances are chosen uniformly spaced in the interval:

$$10^{-6} < \mathrm{Re}\{\beta\} \leq 10^{-1} \tag{3.7}$$

where the lower bound have been slightly increased since if $\mathrm{Re}\{\beta\}$ gets too small, i.e. approximately zero, it will correspond to the case with rigid walls (See Section 2.1.2). Also, the upper bound have been slightly reduced to avoid the $\mathrm{Re}\{\beta\} = 1/8$ but some of the greater values of $\mathrm{Re}\{\beta\}$ still violate the model assumptions.

3.3.5 Structure of Data Sets

In the following, the structure of the data sets in each DSCs are presented. Each of the data sets consists of approximately 10 000 observations, since preliminary experiments showed that this was a reasonable choice to ensure enough training observations.

For the data sets with varying room dimensions and fixed wall admittance the following parameters were chosen; $\mathrm{Re}\{\beta\} = 0.0002$, $N_l = 26$, $N_w = 26$, and $N_h = 15$. Thus, the total number of different rooms is 10 140, which is the number of observations as well since the wall admittance is fixed.

For the data sets with varying wall admittances and fixed room dimensions, the chosen room size is $5 \times 3 \times 2.5\, m^3$. A total number of $N_b = 10\,000$ uniformly spaced real components of the wall admittances were chosen in the interval defined in (3.7).

For the data sets with varying room dimensions and wall admittances the following parameters were chosen; $N_l = 6$, $N_w = 6$, $N_h = 3$, and $N_b = 100$. Thus, there is 108 different rooms and a total of 10 800 observations.

All data sets are structured such that each data set consists of observations from the microphones, and knowledge about the dimensions of the different rooms and wall admittances. An illustration of the data set structure is shown in Figure 3.9. The first tensor[1] `Data` consist of N observations each with L frequency samples from M microphones, the second tensor `Room Dimensions` consist of the room dimension for each observation, and the tensor `Betas` consists of the wall admittances for each observation.

[1] A multidimensional array.

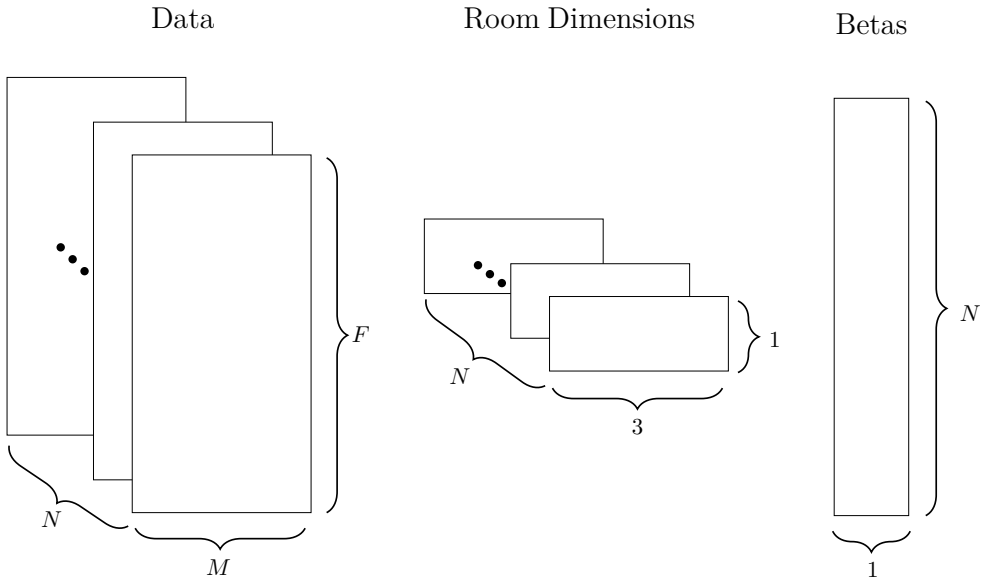


Figure 3.9: Illustration of a data set in a DSC, which consists of three parts. A tensor `Data` containing N observations each with F frequency samples from M microphones, a tensor `Room Dimensions` containing the room dimension for each of the N observations, and a tensor `Betas` containing the wall admittances for each of the N observations

The creation of the DSCs for the training of the CNN models the have now been described. Before presenting the CNN models and the design hereof, an introduction to the general architectures of deep neural networks will be given in the following chapter.

Chapter 4

Deep Neural Networks

In the previous chapter, an introduction to the data generation procedure and a verification of the produced data were given. The generated data will be used to obtain an approximation of the classifier, f^*, which maps the data to a specific class, i.e. $\boldsymbol{y} = f^*(\boldsymbol{x})$ [9, Ch. 6]. *Deep Neural Networks (DNNs)* are often used for defining a mapping $\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\theta})$, where the parameters $\boldsymbol{\theta}$ are optimised to achieve the best approximation of f^* [9, Ch. 6]. In the following, a general introduction to the architectures of DNNs will be given where the main focus will be on the definition of *Convolutional Neural Networks (CNNs)*. The reason for this, that we utilise convolutional layers to extract features from the Room Transfer Functions in the data set collections. Also, convolutional layers have beneficial properties, that we wise to exploit in the models.

How DNNs are trained in general will not been studied here, since the reader should be familiar with this. Instead, we will primarily state the training methods and techniques used in this project.

First, the definition of a *Feed-forward Neural Network (FNN)* and the applied notation will be given, followed by a study on the design of convolutional layers. Next, the definition of residual layers will be given, and we end this chapter by a short description of the chosen loss function and training methods. The chapter will be based on a combination of Goodfellow [9, Ch. 6, 8 & 9], Bishop [3, Ch. 5], Namatevs [21], and He et al. [11].

4.1 Feed-forward Neural Networks

An FNN, also referred to as a multilayer perceptron, is the simplest type of DNN model and can be interpreted as a chain of non-linear functions [9, Ch. 6] [3, Ch. 5]. Let $\boldsymbol{x} \in \mathbb{R}^{N_{in}}$ be a single data sample from a given data set and let $\boldsymbol{y} \in \mathbb{R}^{N_{out}}$ be the desired output, i.e. $\boldsymbol{y} = f^*(\boldsymbol{x})$. An FNN model $f(\cdot; \boldsymbol{\theta})$ is an approximation of f^* and is given as:

$$\boldsymbol{y} \approx f(\cdot;\boldsymbol{\theta}) = (f_L \circ f_{L-1} \circ \cdots \circ f_1)(\boldsymbol{x};\boldsymbol{\theta}) \tag{4.1}$$

where $L \in \mathbb{N}$ is the number of layers, also called the depth of the network [9, Ch. 6] [3, Ch. 5]. The input layer is excluded in the number of layers, which means that implicitly there is an identity layer in (4.1), i.e. $f_0(\boldsymbol{x}) = \boldsymbol{x}$. Each f_l is a non-linear vector function with its own set of trainable parameters [9, Ch. 6]. To get a better understanding of these functions, let us consider the computation of the i'th entry in the output of layer l:

$$z_i^{(l)} = f_l(\boldsymbol{z}^{(l-1)})_i := \varphi^{(l)}\left(\sum_{j=1}^{N_l} w_{i,j}^{(l)} z_j^{(l-1)} + b_i^{(l)}\right) \tag{4.2}$$

where $N_l \in \mathbb{N}$ is the number of nodes in layer l, $w_{i,j}^{(l)} \in \mathbb{R}$ is the weight between node i in layer l and node j in layer $l-1$, $b_i^{(l)} \in \mathbb{R}$ is the bias of node i in layer l, and $\varphi^{(l)} : \mathbb{R} \mapsto \Omega^{(l)}$ is a non-linear function, where $\Omega^{(l)} \subseteq \mathbb{R}$, and is called an *activation function* [9, Ch. 6] [3, Ch. 5]. For the special case $l = 1$ in (4.2), we define $\boldsymbol{z}^{(0)} := \boldsymbol{x}$.

As the name implies the information between layers in an FNN flow only forward in the network [9, Ch. 6] [3, Ch. 5]. Hence, there is no connections where information form the layers are fed back in to the model [9, Ch. 6]. An illustration of a four layer FNN is shown in Figure 4.1, where the arrows illustrates the connections between nodes. We sum up the above in the following definition [9, Ch. 6] [3, Ch. 5].

Definition 4.1 (Feed-forward Neural Network)
Let $L \in \mathbb{N}$ be the number of layers and $N_l \in \mathbb{N}$ be the number of nodes in layer l. Let $\boldsymbol{z}^{(l)} \in \mathbb{R}^{N_l}$ denote the output of layer l, where $\boldsymbol{z}^{(0)} := \boldsymbol{x}$ defines the input. A mapping f is then called an feed-forward neural network if:

$$\boldsymbol{z}^{(L)} = f\left(\boldsymbol{x}; \left\{(\boldsymbol{W}^{(l)}, \boldsymbol{b}^{(l)}) \mid l = 1, 2, \ldots, L\right\}\right)$$

and the output of each layer in f is give as:

$$\boldsymbol{z}^{(l)} = \Phi^{(l)}\left(\boldsymbol{W}^{(l)}\boldsymbol{z}^{(l-1)} + \boldsymbol{b}^{(l)}\right), \quad \text{for } l = 1, 2, \ldots, L$$

where $\boldsymbol{W}^{(l)} \in \mathbb{R}^{N_l \times N_{l-1}}$, $\boldsymbol{b}^{(l)} \in \mathbb{R}^{N_l}$ and $\Phi^{(l)} : \mathbb{R}^{N_l} \mapsto \Omega^{(l)}$, where $\Omega^{(l)} \subseteq \mathbb{R}^{N_l}$. If all entries in $\boldsymbol{W}^{(l)}$ are non-zero for $l = 1, 2, \ldots, L$, then f is said to be *fully connected*.

Remark: Often the vector functions $\Phi^{(l)}$ are point-wise functions, i.e. $\Phi_i^{(l)}(\cdot) = \varphi^{(l)}(\cdot)$ for $i = 1, 2, \ldots, N_l$. These functions are not limited to non-parametrised functions only. Parametrised functions will just increase the parameter set of f. $\Diamond$

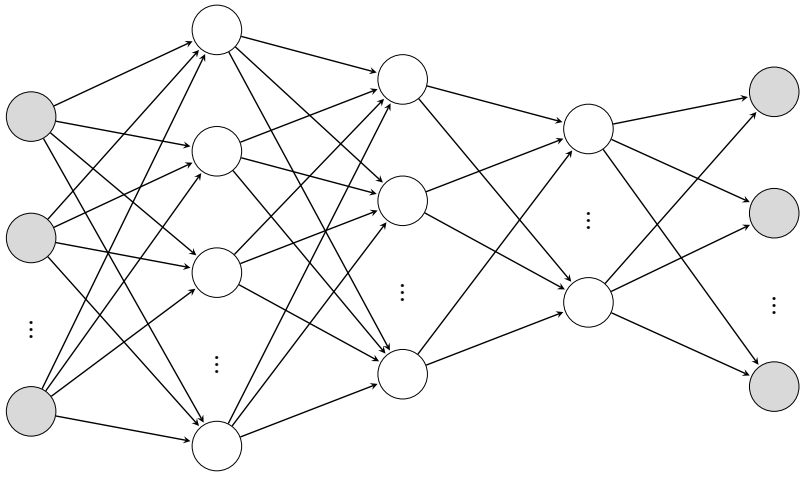


Figure 4.1: Illustration of a four layer FNN, where each arrow corresponds to a weighted connection between each node.

The FNN model architecture is utilised in the CNN models for classifying the extracted features of the input to the models. In order to design an optimal architecture, there are no theoretical rules or guidelines but must be obtained through experiments [9, Section 6.4]. However, the choice of activation functions depend on the type of layer, i.e. if it is a hidden or an output layer. Both theoretical aspects and experiments underlie the choices of activation functions, which will be presented in the following.

4.1.1 Activation Functions

Various activation functions are described in the literature and it would be too comprehensive to state all of them here [9, Ch. 6] [3, Ch. 5]. Thus, we only state the most commonly used activation functions, which will be used for designing the CNN models in Section 5.1.

Rectified Linear Unit

One of the most used activation functions, or generalisation hereof, is the *Rectified Linear Unit (ReLU)* [9, Sec. 6.3.1]:

$$\phi(z) = \max\{0, z\} \tag{4.3}$$

The ReLU activation function has become the default recommendation for hidden layers in modern DNNs [10, Sec. 6.1]. The reason for this is that the ReLU function makes the weights and biases in a layer easy to optimise due to ReLU being very similar to a linear activation [10, Sec. 6.3.1]. During training of a model (See Section 4.4), the derivatives remain large through a ReLU activation due to its derivative

being one for $z \geq 0$ and zero else. Also, the second order derivative is zero almost everywhere, which means that the gradient direction is far more useful for training the model than an activation function that introduce second-order effects [10, Sec. 6.3.1].

However, one drawback to the ReLU activation function is that the node weights and biases cannot be optimised via gradient-based methods on training examples where $z < 0$, i.e. zero-gradient [10, Sec. 6.3.1]. Thus, in order to avoid zero-gradients, different generalisation of the ReLU activations function have been proposed. One of these are the *Leaky Rectified Linear Unit (LeakyReLU)* [10, Sec. 6.3.1]:

$$\phi(z) = \max\{\alpha z, z\}, \quad \alpha \in \mathbb{R} \tag{4.4}$$

The parameter α is fixed and often chosen to be a small value, e.g. $\alpha = 0.01$ [10, Sec. 6.3.1]. The α parameter can also be treated as a trainable parameter and in this case the activation function in (4.4) is called the *Parametric Rectified Linear Unit (PReLU)* function [10, Sec. 6.3.1]. A plot of ReLU and LeakyReLU are shown in Figure 4.2a and Figure 4.2b, respectively.

Sigmoid and Softmax

The choice of activation function in the output layer is heavily dependent on the task at hand. For classification problems, there is two functions that are often used, depending on the number of classes. The first activation function is the *sigmoid* function, also called the *logistic sigmoid* function [10, Sec. 6.2.2.2] [3, Sec. 4.2]:

$$\phi(z) = \sigma(z) := \frac{1}{1 + e^{-z}} \tag{4.5}$$

The sigmoid function is used for two-class classification problems and only requires a single node in the output layer. The sigmoid function can be interpreted as a transformation of z into a probability [10, Sec. 6.2.2.2]. This follows if the DNN is considered as a conditional probability distribution. Considering two classes, i.e. $\mathcal{C}_1$ and $\mathcal{C}_2$ then the posterior probability of $\mathcal{C}_1$ is given as [3, Sec. 4.2]:

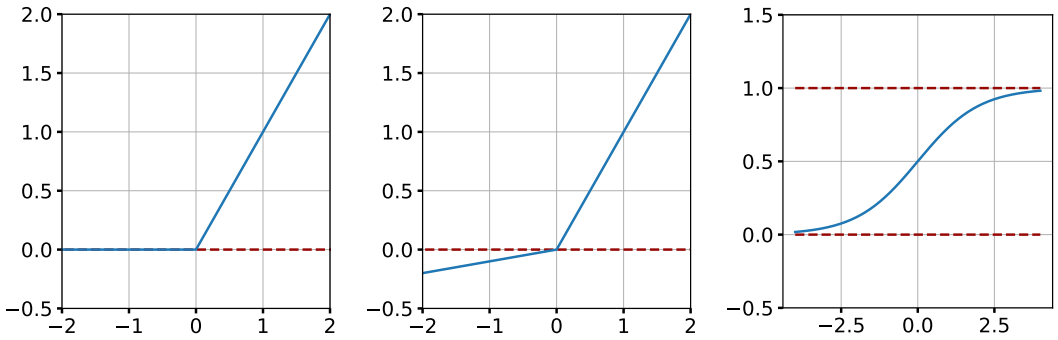
$$\begin{aligned} P(\mathcal{C}_1 \mid \boldsymbol{x}) &= \frac{P(\boldsymbol{x} \mid \mathcal{C}_1)P(\mathcal{C}_1)}{P(\boldsymbol{x} \mid \mathcal{C}_1)P(\mathcal{C}_1) + P(\boldsymbol{x} \mid \mathcal{C}_2)P(\mathcal{C}_2)} \\ &= \frac{1}{1 + e^{-a}} = \sigma(a), \\ a &:= \log\left(\frac{P(\boldsymbol{x} \mid \mathcal{C}_1)P(\mathcal{C}_1)}{P(\boldsymbol{x} \mid \mathcal{C}_2)P(\mathcal{C}_2)}\right) \end{aligned}$$

where $\boldsymbol{x}$ is the input of the network. It is easily seen from the above that $\sigma(-a) = 1 - \sigma(a)$, and therefore $P(\mathcal{C}_2 \mid \boldsymbol{x}) = 1 - P(\mathcal{C}_1 \mid \boldsymbol{x})$. Thus, the output of a layer with a sigmoid activation function will follow a Bernoulli distribution [10, Sec. 6.2.2.2] [3, Sec. 4.2]. The sigmoid function is depicted in Figure 4.2c.

For classification problems with more than two classes, there is a similar activation to the sigmoid function for two-classes, but this function requires a node for each class, a one-hot encoding, in the output layer. This activation functions is the *softmax* function [10, Sec. 6.2.2.3] [3, Sec. 4.2]:

$$\Phi(\boldsymbol{z})_i = \frac{\exp(z_i)}{\sum_{j=1}^{K} \exp(z_j)}, \quad \text{for } i = 1, 2, \ldots, K \tag{4.6}$$

where K is the number of classes. Similarly to the sigmoid function, softmax can be interpreted as a transformation of $\boldsymbol{z}$ into a discrete probability distribution [10, Sec. 6.2.2.3]. Following the same steps as above, it can be shown that the output of a layer with the softmax activation function follows a Multinoulli distribution [10, Sec. 6.2.2.3] [3, Sec. 4.2]. The softmax activation function is difficult to illustrate due to its high dimensionality, but it resembles a sigmoid activation function in greater dimensions.



(a) ReLU **(b)** LeakyReLU **(c)** Sigmoid

Figure 4.2: Plots of three activation functions; (**a**) ReLU, (**b**) LeakyReLU with $\alpha = 0.1$, and (**c**) Sigmoid.

4.2 Convolutional Neural Networks

A CNN is, as the name implies, a neural network which has at least one convolutional layer. This allows CNNs to compactly represent highly non-linear and varying functions [21]. The current architecture paradigm for CNNs consists of two parts. The first part consists of repeated stages with convolution, non-linear activation and pooling [9, Sec. 9.2 & 9.3] [21]. The second part consists of at least one fully connected layer as in an FNN [21]. In this section, we will give a short introduction to the construction of convolutional layers, which are used for the first part of a CNN[1].

There are different interpretations of the structure of a convolutional layer in the literature [9, Sec. 9.3]. E.g. one terminology is that the convolutional, non-linear

[1]For further details we refer the reader to [9, Ch.9]

activation, and pooling are defined as a single layer [9, Sec. 9.3]. In this project the chosen terminology interpret convolution and polling as separated layers where the non-linear activation is included in the convolutional layer. The reason for this is to reflect the syntax of the Python packages Tensorflow and Keras. Using this terminology a single fully connected layer, as given in Definition 4.1, is called a *Dense* layer. Henceforth, we will use this terminology to properly distinguish between layers.

A convolutional layer utilises the discrete convolution operator instead of matrix multiplication as in dense layers [9, Sec. 9.2]:

$$y_n = (\boldsymbol{x} * \boldsymbol{w})_n = \sum_{m=-\infty}^{\infty} x_m w_{n-m} \tag{4.7}$$

where $\boldsymbol{x} = \{x_n\}_{n\in\mathbb{Z}}$ and $w = \{w_n\}_{n\in\mathbb{Z}}$ are two real sequences [9, Sec. 9.1]. In practice, $\boldsymbol{x}$ and $\boldsymbol{w}$ will both be limited to have support on a finite index set, e.g. $x_i \neq 0$ for any $i \in \mathcal{I}$ with $|\mathcal{I}| < \infty$. Hence, both sequences can be interpreted as vectors, i.e. $\boldsymbol{x} \in \mathbb{R}^{N_1}$ and $\boldsymbol{w} \in \mathbb{R}^{N_2}$ [9, Sec. 9.1].

Instead of nodes as in a dense layer, a convolutional layer is specified by a number of filters and a filter size, also called *kernels* and *kernel size* respectively [9, Sec. 9.1 & 9.2]. Thus, every kernel in the same layer have equal sizes, and these kernels are the trainable parameters of the layer. Similarly to dense layers, the activation function is a point-wise function and can be parametric as well (See Section 4.1.1). The definition of a convolutional layer is stated in the following [9, Sec. 9.1 & 9.2]:

Definition 4.2 (Convolutional Layer)
Let $N_K \in \mathbb{N}$ be the kernel size and $K \in \mathbb{N}$ be the number of kernels. Let $\boldsymbol{X} \in \mathbb{R}^{N\times M}$ denote the input and $\boldsymbol{Y} \in \mathbb{R}^{\tilde{N}\times N_K}$ denote the output, where $N, M \in \mathbb{N}$, $\tilde{N} = N - N_K + 1$ and $N \geq N_K$. Then, the j'th column in $\boldsymbol{Y}$ is given by:

$$\boldsymbol{y}_j := \Phi\left(\sum_{i=1}^{M} \boldsymbol{w}_{i,j} * \boldsymbol{x}_j + \boldsymbol{b}_j\right)$$

where $\boldsymbol{W_j} := [\boldsymbol{w}_{1,j}, \boldsymbol{w}_{2,j}, \ldots, \boldsymbol{w}_{M,j}]$ is the j'th kernel, $\boldsymbol{b}_j$ is the j'th bias, and $\Phi : \mathbb{R}^{\tilde{N}\times N_K} \mapsto \mathbb{R}^{\tilde{N}\times N_K}$ is a non-linear activation function.

In the literature, the output of a convolutional layer is referred to as a *feature map* or a *kernel map* [21] [9, Sec. 9.2]. Thus, the network is said to learn different features of the training data. The first convolutional layer will extract low-level meaningful features, e.g. edges or corners of a picture using 2D-convolution [21]. The following convolutional layers in a CNN will extract higher-level features and so forth[21]. An illustration of a convolutional layer with a single kernel of size three is shown in

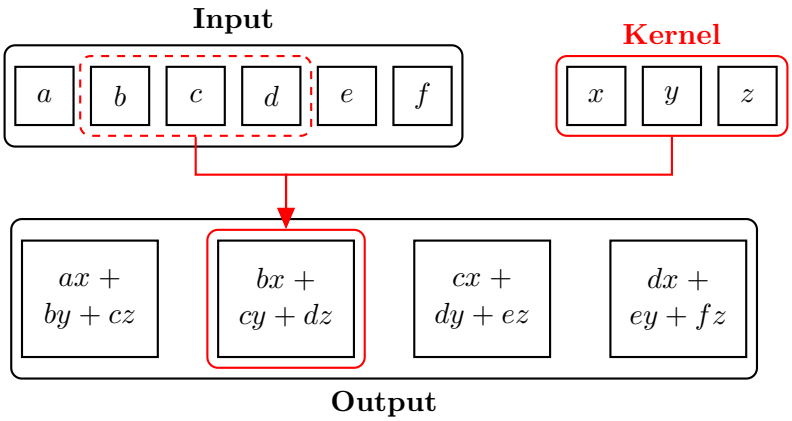


Figure 4.3: Illustration of a convolutional layer where the non-linear activation have been included in the "Output" block.

Figure 4.3. Note that in many implementations of convolutional layers, it is possible to control the sliding of the kernels [21]. This is referred to as *strides* and in Figure 4.3 the number of strides is one, as for standard discrete convolution (See Equation (4.7)). From Figure 4.3, it can be seen that the end points of the input data are only used once in the computation of the output, i.e. entry a and f in Figure 4.3. Depending on the data used for training, the end points might contain important information. Therefore, most implemented convolutional layers do zero-padding at the ends of the input such that in Definition 4.2, the output size becomes $\tilde{N} = N$ [21].

In the history of DNNs, CNNs have shown a great impact on various applications in the field [9, Sec. 9.11]. As mentioned in Chapter 1, many of the state-of-the-art DNN architectures are CNNs. The reason for this is that convolutional layers exploit three ideas for improving machine learning systems in general; sparse interactions, parameter sharing, and equivariant representations [9, Sec. 9.2]. This is why we choose to use CNN models for solving of the classification problem to begin with.

Sparse interactions, also called sparse weights or sparse connectivity, refer to how a single input unit and a single output unit of a layer interacts with each other [9, Sec. 9.2]. For a standard dense layer, every input unit are used for computing a single output unit due to matrix multiplication (See Definition 4.1). In convolutional layers, however, only a small number of input units are used for computing the output unit, as illustrated in Figure 4.3 [9, Sec. 9.2]. In the figure, only three input units are used at the time for computing the output as the kernel slides over the input. This reduces the number of parameters required in the model and fewer operations are required to compute the output [9, Sec. 9.2]. It also improves the statistical efficiency of the model [9, Sec. 9.2].

Parameter sharing refers to the parameters being used multiple times [9, Sec. 9.2],

as illustrated in Figure 4.3 where the same kernel is used to compute the output. In a dense layer, each weight in the weight matrix will only be used once for computing the output. Again, this reduces the number of parameters in the model and improves the statistical efficiency [9, Sec. 9.2]. A consequence of the shared parameters is that the layer have the property of equivariant representations [9, Sec. 9.2]. This means that if the layer input changes then the output changes in the same way as the input [9, Sec. 9.2]. However, it is not all transformations of the input this apply to. Convolutional layers are only equivariant to translations of the input and not rotations of the input [9, Sec. 9.2].

4.2.1 Pooling Layers

It has become mandatory to follow up a convolutional layer with a pooling layer, since from experience it has shown to increase performance and make the output representation approximately invariant to small translations of the input [21] [9, Sec. 9.3]. A pooling layer applies a *pooling function* on a window of the input at a time [21] [9, Sec. 9.3]. Hence, one can think of a pooling layer as a kind of convolution layer with a single and known kernel. Just as for convolutional layers, one can control the sliding of the window by specifying the number of strides. However, the standard number of strides is equal to the window size such there is no overlap [9, Sec. 9.3].

The two most common pooling functions are *Maximum Pooling (MP)* and *Average Pooling (AP)* [21] [9, Sec. 9.3]:

$$f_{MP}(x) = \max_{i \in [N_w]} x_i \tag{4.8}$$

$$f_{AP}(x) = \frac{1}{N_w} \sum_{i=1}^{N_w} x_i \tag{4.9}$$

where $N_w \in \mathbb{N}$ is the window size of the pooling, $[N_w] := \{1, 2, \cdots, N_w\}$ and x is the windowed input. A pooling layer thereby reduces the dimensionality of the input. Let N be the size of the input and N_w the window size. Then the output size of the pooling layer, N_P, without zero-padding and a stride equal to N_w is given by:

$$N_P = \left\lfloor \frac{N}{N_w} \right\rfloor \tag{4.10}$$

In the literature, the pooling window and window size are called the *pooling kernel* and *pooling size* respectively [9, Sec. 9.3].

4.3 Residual Layers

In recent years, a degradation problem have been exposed in the study of DNNs; as the number of layers increases in the model, the accuracy gets saturated and begin to degrade rapidly [11]. A proposed solution by He et al. [11] is to use *residual layers*,

also call *residual conncetions*. It may be difficult for a block of layers in a DNN to learn a desired underlying mapping g from the training data. Hence, the main idea behind a residual layer is to force the block of layers to learn the residual mapping instead [11]:

$$f(\boldsymbol{z}) := g(\boldsymbol{z}) - \boldsymbol{z} \tag{4.11}$$

In practise, a residual layer is simply implemented by making a short-cut connection [11], as illustrated in Figure 4.4. The whole block of layers and the short-cut connection is referred to as a residual layer. In some cases, the DNN could learn to skip the layers entirely and the underlying mapping would be an identity mapping, i.e. $g(\boldsymbol{z}) = \boldsymbol{z}$ [11]. Depending on the choice of layers between the short-cut connec-

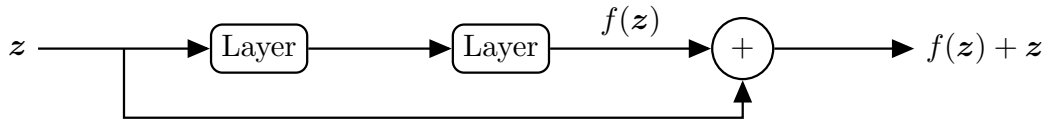


Figure 4.4: Simple illustration of a residual connection.

tion, some sort of up-sampling or down-sampling of the input might be necessary to match the output dimensions of the layers [11]. For example if both layers are convolutional layers then both layers must have the same number of kernels and the same kernel size [11]. Also, the input should be "up-sampled" in the short-cut connection using a convolutional layer with the same number of kernels as the other layers but a kernel size of one [11].

We would like to utilise residual layers in the architecture of the CNN models because they have shown to improve the performance for very deep models. Also, since we would not be focusing on optimising the number of layers in this project, residual layers make it easy to design models with high performance. For example, if we choose too many layers, the model would learn to used the skip connections only.

4.4 Training with Supervised Learning

Now that some of the building blocks of DNNs have been presented in the previous sections, we will give a short introduction to the training of DNNs. The models considered in this project (See Section 5.1) are trained using *supervised learning*, which means that observations of the ground truth are available to guide the training [10, Sec. 8.1]. Henceforth, let $\boldsymbol{y}$ denote the ground truth, also called the label, of input sample $\boldsymbol{x}$ and let $\hat{\boldsymbol{y}}(\boldsymbol{x}, \boldsymbol{\theta})$ denote the estimated label obtained from a DNN model, where $\boldsymbol{\theta}$ is the parameters of the model.

Since the problem considered in this project is a classification problem with multiple classes (See Section 1.3), the ground truth will be a single class label for each input. Thus, the softmax activation function will be utilised in the output layer

(See Section 4.1.1). The softmax activation function returns a discrete probability distribution where the entry with greatest probability is the predicted class label (See Section 1.3). Thus, the ground truth distribution is given as a so called *one-hot vector*:

Definition 4.3 (One-Hot Vector)
Let $K \in \mathbb{N}$ and $\boldsymbol{e}_j := [e_{j,1}\ e_{j,2}\ \cdots\ e_{j,K}]^\intercal \in \mathbb{R}^K$. Then, $\boldsymbol{e}_j$ is called a one-hot vector if:

$$e_{j,i} = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases}$$

Given a set of predicted labels, $\hat{\boldsymbol{Y}} = \{\hat{\boldsymbol{y}}(\boldsymbol{x}_1, \boldsymbol{\theta})_1, \hat{\boldsymbol{y}}(\boldsymbol{x}_2, \boldsymbol{\theta})_2, \ldots, \hat{\boldsymbol{y}}(\boldsymbol{x}_N, \boldsymbol{\theta})_N\}$ and a set of corresponding ground truth labels, $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N\}$, then a DNN for multiple-class classification is trained by solving the following optimisation problem [10, Sec. 8.1] [3, Sec. 5.2]:

$$\boldsymbol{\theta}^\star := \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \tag{4.12a}$$

$$= \arg\min_{\boldsymbol{\theta}} \frac{1}{NK} \sum_{i=1}^{N} \sum_{k=1}^{K} y_{i,k} \log\left(\hat{y}_{i,k}(\boldsymbol{x}_i, \boldsymbol{\theta})\right) \tag{4.12b}$$

where $\mathcal{L}$ is called a *loss-function* [10, Sec. 8.1]. For this specific classification problem, the appropriate choice of loss-function is the crossentropy, also known as the categorical crossentropy, as given in (4.12b). In general, the choice of loss-function is problem dependent, but we will not go further into studying the different loss-functions. Different algorithms for solving the optimisation problem in (4.12a) exist in the literature [10, Ch. 8]. These algorithms are gradient based and in order to compute the gradients, they utilise the backpropagation algorithm [10, Sec. 8.3] [3, Ch. 5.3]. We will not go into further details, but instead refer the reader to Goodfellow [10, Ch. 8] and Bishop [3, Ch. 5] for more on the topic of training DNNs. Later when we present the design of our CNN models, we will specify the choice of training algorithm as well.

In this chapter, the theory on the architecture on DNNs and how they are trained have been given. In the following chapter, we will present the design of our CNN models and how the training data have been preprocessed. Also, we will state the experiments that have been conducted to validate the performance of the CNN models.

Chapter 5

Models and Simulation Experiments

In the previous chapter, the necessary theory on Deep Neural Networks (DNNs) were introduced, which will be used designing the Convolutional Neural Network (CNN) models in the following chapter. First, the CNN models will be presented followed by a description of the conducted simulation experiments. We end this chapter with explaining the preprocessing of the data sets.

5.1 CNN Architectures

In Section 1.3, the general architecture of the CNN models were given. The CNN models will take the magnitude response of M RTFs as input to the network, where M is the number of microphones in the room. All of our generated data sets use the same fixed parameters to sample the frequency axis as described in Section 3.3. Thus, the input is given as a matrix with shape $855 \times M$ (See Section 3.3). The output of the CNN models is given as a one-hot vector (see Section 4.4), which is the preferred approach when dealing with multi-class classification. Thus, the output dimension is solely depending on the number of classes. The number of different classes and how they are defined for our experiments will be presented in Chapter 6.

As already mentioned in Section 1.3, the CNN architectures are inspired by some of the ideas used in WaveNet [28] and U-Net [23], which we will explain a bit further in the following. All of our CNN models follow the same general architecture but vary in number of layers, number of nodes in the dense layers, etc. Thus, we will describe the general architecture and end the section by presenting the specific architecture of our models.

The general architecture consists of two parts; a feature extraction part and a feed-forward part. The feed-forward part is basically a fully connected feed-forward network (See Definition 4.1), which takes the features from the feature extraction and

return a discrete probability distribution (See Section 1.3.3 and 4.1.1). The number of layers and nodes in the feed-forward part varies depending on the task at hand. Each layer in the feed-forward part uses ReLU (See Section 4.1.1) as activation function, except from the output layer, which uses a Softmax activation (See Section 4.1.1). The Softmax activation is essential, since it returns the probabilities of the input belonging to the different classes.

The feature extraction part consist of a sequence of convolutional and pooling layers (See Definition 4.2 and Section 4.2.1) inspired by the first half of the U-Net [23] architecture. This first half of U-Net [23] follows a specific sequence of layers; two convolutional layers with the same number of filters and ReLU activation, followed by a maxpooling layer, and the pattern is repeated while increasing the number of filters each time. Inspired by this architecture, we have created a *Convolution Residual (CRes) block* that consists of two convolutional layers and a pooling layer, in order to make the network design simple. As the name implies the CRes block also got a residual connection (See Section 4.3), which was inspired by how WaveNet [28] utilises residual connections in each block of the network. An illustration of the CRes block is shown in Figure 5.1.

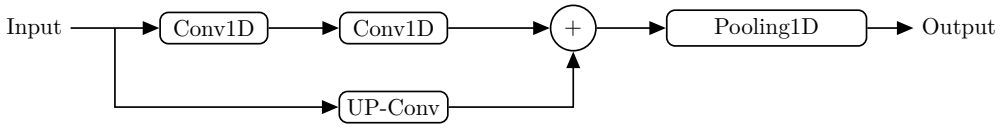


Figure 5.1: Illustration of a single CRes block. A single block consists of two 1D-Convolutional layers (*Conv1D*) both with a LeakyReLU activation. The residual connection uses a 1D-Convolutional layer (*UP-Conv*) to match the number of features. The addition layer is followed by a 1D-Pooling layer (*Pooling1D*).

The *UP-Conv* is a 1D-Convolutional layer with a fixed filter size of 1, which is used to "up-sample" the input such that the number of channels of the residual connection matches the number of output filters of the second *Conv1D* layer (See Section 4.3). Each convolutional layer in a CRes block have the same number of filters, using padding to keep the dimension of each channel and is followed by a LeakyReLU activation (See Section 4.1.1). From preliminary experiments, and using the knowledge from both U-Net [23] and WaveNet [28], we fix some of the parameters for the CRes block. These parameters and their value are given in Table 5.1. This means that the only free parameters for a CRes block are the number of filters and the pooling type.

Filter Size	Filter Strides	Pool Size	Pool Strides
3	1	2	2

Table 5.1: Fixed parameters of the CRes block (See Chapter 4 for a definition of the terms).

Now that we have described the general network architecture, we present the specific architectures for the two CNN models. The first CNN model is for room volume classification given in Table 5.2 and the second is for wall admittance classification given in Table 5.3. The feature extraction part and the feed-forward part are connected through a flatten layer, which stacks the features of the last CRes block into a vector. An illustration of the CNN model for room volume classification is depicted in Figure 5.2.

Feature Extraction Part						
Block	CRes(1)	CRes(2)	CRes(3)	CRes(4)	-	-
Filters	64	128	256	512	-	-
Pooling Type	Average1D	Average1D	Average1D	Max1D	-	-
Feed-Forward Part						
Layer	Dense(5)	Dense(6)	Dense(7)	Dense(8)	Dense(9)	Output(10)
Nodes	1800	1024	512	256	128	#Classes
Activation	ReLU	ReLU	ReLU	ReLU	ReLU	Softmax

Table 5.2: An overview of the room volume classifier architecture. The numbers in parenthesis indicates the order, that the blocks and layers are connected.

Feature Extraction Part							
Block	CRes(1)	CRes(2)	CRes(3)	CRes(4)	CRes(5)	-	-
Filters	64	128	256	512	1024	-	-
Pooling Type	Max1D	Max1D	Max1D	Max1D	Max1D	-	-
Feed-Forward Part							
Layer	Dense(6)	Dense(7)	Dense(8)	Dense(9)	Dense(10)	Dense(11)	Output(12)
Nodes	3600	1800	1024	512	256	128	#Classes
Activation	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU	Softmax

Table 5.3: An overview of the wall admittance classifier architecture. The numbers in parenthesis indicates the order that the blocks and layers are connected.

All of our CNN models are trained using supervised learning with categorical cross-entropy (See Section 4.4) as the loss function. This loss function is the most applied for multi-class classification. Regarding the choice of optimisation method, we have examined both *Stochastic Gradient Descent (SGD)* [10, Sec. 8.3.1] and *Adam* [14], and have chosen to use SGD. The reason for this is that despite Adam is faster, it turned out to be unstable. We have trained our models for a few of our data sets using both optimisation methods to examine their performance. In Figure 5.3, the training loss for the wall admittance classifier (See Table 5.3) from three different training runs is depicted. In order to compare the training losses for different optimisation methods, we have used the same initialisation of the model for both methods as well as keeping the same division of the data set into training and test data.

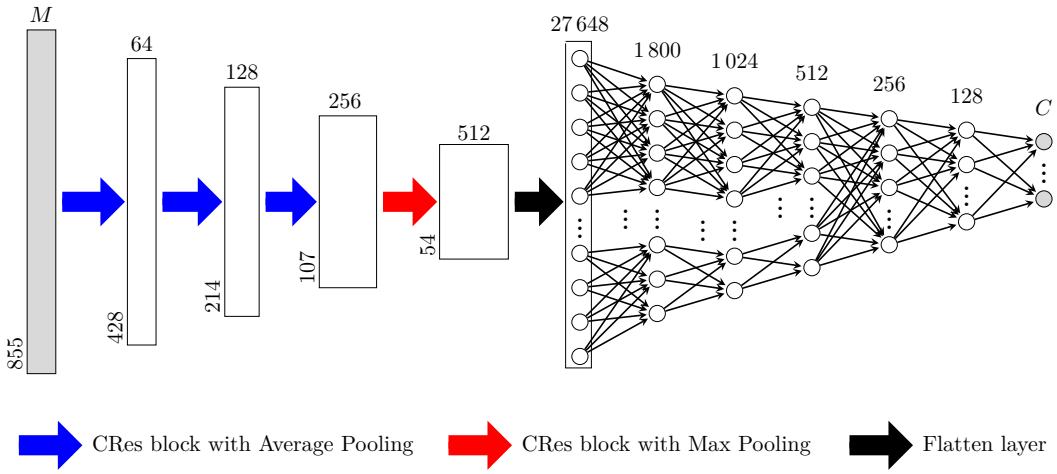


Figure 5.2: Illustration of the CNN model for room volume classification. The grey toned box illustrates a single input observation, with M channels and 855 frequency samples. The white boxes illustrate the output dimensions for each of the CRes blocks, where the value on top is the number of filters and the value on the side is the number of samples. The circles illustrates the nodes of the dense layers, where the value above is the total number of nodes in the give layer. The grey toned nodes is the output layer of the network.

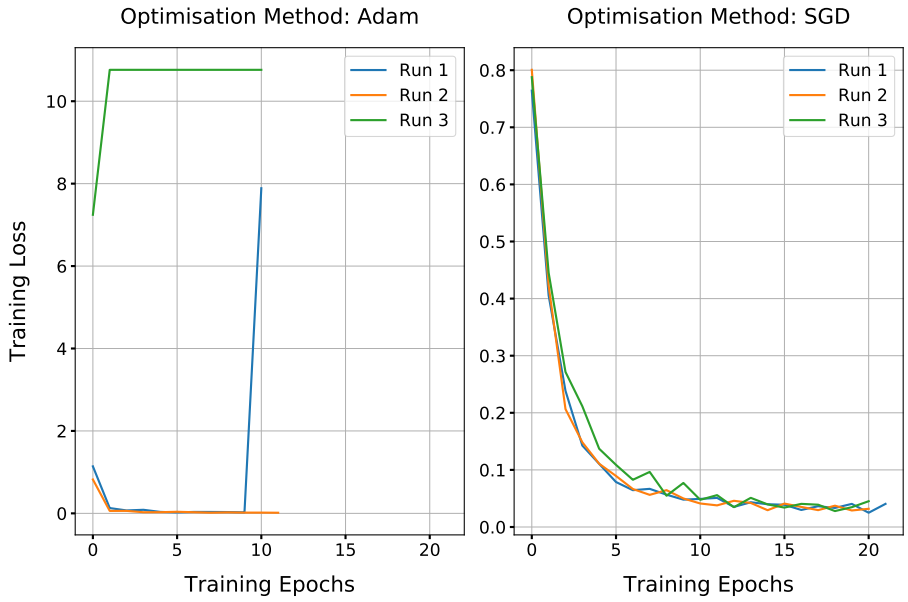


Figure 5.3: Training loss for classifying the wall admittance using Adam(**left**) and SGD(**right**) for the training. Number of microphones is 64, the microphone grid is centred in the middle of the room, and total number of classes is 3.

For each run, a different split of the data set is used and different initial weights in the model. The data set used to obtain Figure 5.3 is the data set with varying wall

admittances and fixed room dimensions where 64 microphones are placed in a cubic grid centred in the middle of the room (See Section 3.3) and three classes were used. From Figure 5.3, it is evident that Adam tends to be unstable compared to SGD. The training losses converge as expected using SGD but this is not the case for Adam. Using Adam the training loss converges nicely at first but starts to diverge after a few epochs, and sometimes the training loss diverges from the very beginning. This was a general problem that kept appearing when training different models or using different data sets. It is hard to tell what makes the training loss converge or diverge using Adam, since it seems to happen at random and the parameter optimization problem is not convex. Thus, SGD seems as the best choice for training our models.

5.2 Description of Simulation Experiments

Different simulation experiments have been conducted to study the accuracy performance of the different CNN models described above. From here on out, we will refer to these as just experiments. In general, two different experiment set-ups were used; one for studying the accuracy performance for different number of classes, and one studying the robustness against additive noise. Since the Data Set Collections (DSCs) contain data sets for different microphones configurations and generation set-ups (See Section 3.3), the experiments will be designed such that the CNN models are trained using all of the data sets in each DSC, one at a time. The results of these experiments will be presented in Chapter 6.

5.2.1 Experiment for Classification Performance

The first experiment examine the classification performance of the CNN models for different number of classes against the number of microphones. The experiment is repeated for both of the CNN architectures described in the above section. For a given DSC and choice of CNN model, the experiment is conducted as follows.

For each of the data sets in the DSC, a CNN model is initialised for a given number of classes. Hence, a new model are considered for each number of classes, since this changes the output dimension of the CNN models. The different number of classes have been chosen, such that there is equivalent number of observations in each class. If the number of observations in each class are heavily unbalanced, the CNN model could end up learning to classify all observations as the class with the majority. For room volume classification, we will examine the performance for $C_V \in \{2, 3, 7, 10, 20\}$ classes, and for wall admittance classification $C_\beta \in \{2, 3, 5, 7, 10\}$ classes are chosen to examine the performance. The choice of these particular number of classes are based on a preliminary analysis of the data sets (See Chapter 6).

When a number of classes have been chosen, the CNN model can be trained. In order to get a better idea of the models classification performance in general, we

utilise k-fold cross validation. With this, the data set is divided into k folds where one fold at a time are used as the validation set and the remaining folds is used as the training set. When a training and a validation set are given, the CNN model is initialised and trained with the training set. After training, the CNN model is applied to the validation set and the classification accuracy is saved. Then, the process are repeated for the next folds, until the CNN model have be been initialised and trained a total of k times. The average classification accuracy and standard deviation are computed, which we use to measure the classification performance of the model for the given the microphone configuration, number of microphones, and number of classes (See Section 6.1 for further description). The whole process is repeated for all of the different number of classes and data sets.

All of the CNN models are trained using the SGD as mentioned in Section 5.1. From preliminary experiments, we found that the training loss converges faster for a smaller number of classes. Hence, we have chosen to utilise early stopping to prevent the models from over-fitting. The early stopping procedure monitors the validation accuracy during the training of the model. The training stops if there is no improvement in the accuracy over the last 2 epochs, and the best parameters in terms of accuracy are returned. The maximum number of trainings epochs is fixed to 50 and the batchsize is fixed to 150 observations, based on the results from preliminary experiments.

For a given DSC, we can compare classification performance of the CNN models for different number of microphones and classes. The CNN models can also be compared across the DSCs and thereby examine how the different microphone configurations and generation set-ups of the rooms affect the performance.

5.2.2 Experiment with Additive Noise

This experiment examines how robust the trained CNN models are against additive noise by measuring the classification accuracy over noisy RTFs. The design of this experiment are the same as the one described above except from generated complex Gaussian noise are added to the validation sets. Thus, we will only describe how the noise are added to the data.

Before the data is normalised as described in Section 5.3, a complex Gaussian noise signal are added to the RTFs. The complex Gaussian noise signal has zero mean and the variance are computed from the signal-to-noise ratio (SNR). Let $\boldsymbol{H} \in \mathbb{C}^N$ be the RTF for a single microphone in a single observation and let σ^2 be the variance of the complex Gaussian noise signal. Then, the SNR in dB is given as:

$$\mathrm{SNR}_{dB} = 20 \log_{10} \left(\frac{N^{-1/2} \left\| \boldsymbol{H} \right\|_2}{\sigma} \right) \tag{5.1}$$

Isolating the standard deviation σ in 5.1 and one obtains the following:

$$\sigma = N^{-1/2} \left\| \boldsymbol{H} \right\|_2 \cdot 10^{\frac{\text{SNR}_{dB}}{20}} \tag{5.2}$$

The complex Gaussian noise signal $\boldsymbol{v} \in \mathbb{C}^N$ is generated using two Gaussian processes $\boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathbb{R}^N$ with zero mean and standard deviation $\sigma/\sqrt{2}$. Thus, the complex Gaussian noise signal is computed as $\boldsymbol{v} = \boldsymbol{u}_1 + j\boldsymbol{u}_2$, such that the desired SNR is obtained for the experiment. This process is repeated for all of the observations in a data set, but only during the validation of the model. The means that the CNN models are trained without additive noise and evaluated using the validation set with added noise. An illustration of a RTF without noise and the same RTF with additive complex Gaussian noise corresponding to 0dB SNR is shown in Figure 5.4.

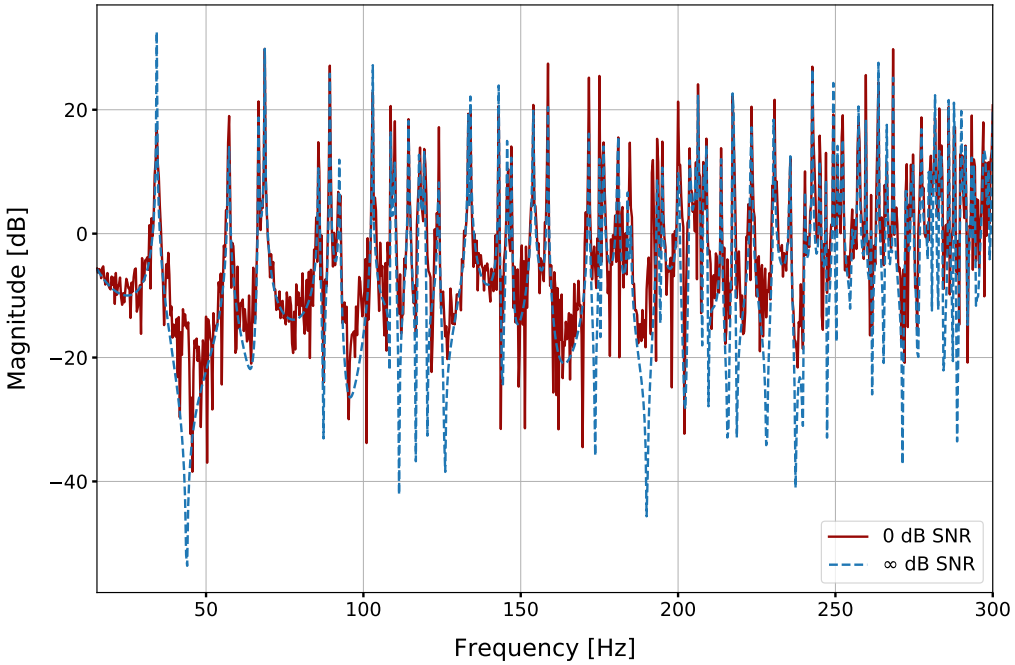


Figure 5.4: An illustration of (**blue**) an arbitrary RTF without noise for a room with size $5\times3\times2\,m^3$ and (**red**) the same RTF with additive complex Gaussian noise corresponding to 0dB SNR.

The experiment is repeated where the CNN models are trained with additive noise as well. However, the SNR for the training sets are fixed to 15dB independent of the SNR for the validation sets.

5.3 Preprocessing

The data set consists of RTFs, which are complex due to the Fourier transform (See Chapter 2). The RTFs are stored in a tensor $\boldsymbol{H}$, and before the data can be used

for the training it must be preprocessed, since CNNs in general do not take complex values as input as mentioned in Section 1.3. We have chosen to preprocess the data by taking the log-magnitude responses as follows:

$$\hat{H}_{n,m,f} = \log_{10} |H_{n,m,f}|, \quad (n,m,f) \in \mathcal{I} \tag{5.3}$$

where $\mathcal{I} := [N] \times [M] \times [F]$, $[N] := \{0, 1, \ldots, N-1\}$, and $N, M, F \in \mathbb{N}$ are the dimensions of the data tensors (See Section 3.3.5). The RTFs are also normalised before given as input to the CNN models. The normalisation is done by dividing the whole data set by the maximum absolute log-magnitude over the whole set:

$$\gamma := \max_{(n,m,f)\in\mathcal{I}} \left|\hat{H}_{n,m,f}\right| \tag{5.4}$$

$$\tilde{H}_{n,m,f} = \gamma^{-1} \hat{H}_{n,m,f}, \quad (n,m,f) \in \mathcal{I} \tag{5.5}$$

By scaling the log-magnitude responses by the same constant, we ensure that the information regarding the magnitude of the resonance frequencies is maintained. This is important since we expect the locations of the resonance frequencies in the magnitude response to help the CNN model identifying the room volume, and their magnitudes are necessary for determining the wall admittances (See Section 1.3.2). If we instead normalised the data by subtracting the mean and divide by the standard deviation, we would "distort" the magnitude responses and thereby make it harder for the CNN models to do classification. Thus, the suggested normalisation in (5.5) is the most reasonable choice such that no important information is lost and still normalising the data as suggested for any kind of DNNs [10, Section 8.7].

Furthermore, since we only store the dimensions of the rooms in the data sets, the room volume must be calculated in order to do room volume classification. This is fairly easily done by computing the product of the room dimensions for each room. However, since we are using supervised learning with one-hot vectors (See Section 4.4) as target output distributions, we must compute the one-hot vectors for both the room volume and the wall admittance classes. In the following chapter, the classification results for the different experiments will be presented.

Chapter 6

Results

In the following chapter, the main results of the room volume and wall admittance classification for the simulation experiments described in Section 5.2 will be presented. All tables of the results from the experiments are given in Appendix C where only some of them are shown in the following sections.

Before presenting the results of our different classification experiments, we give a description of the performance measure, which will be used to evaluate the classification.

6.1 Performance measure

In this section, we will present how we evaluate the classification performance. We use the accuracy to measure the classification performance of a trained network since it is the most used performance measurement for classification problems. The accuracy states how well a CNN model classifies the observations given as the percentage of correctly classified observations:

$$\mathbb{A} = \frac{N_C}{N}$$

where N_C is the number of correctly classified observations and N is the total number of classified observations. Since the performance of a CNN model depends on the initialisation of its weights, we have chosen to use k-fold cross validation to get the general picture of the classification performance (See Section 5.2). An illustration of the data set division for the k-fold cross validation is shown in Figure 6.1. For each of the k folds, a k'th part of the data set is used as validation set and the remainder of the data set is used for training. A total number of 10 folds have been chosen, e.g. for a data set with 10 000 observations each fold contains a validation set of 1 000 observations.

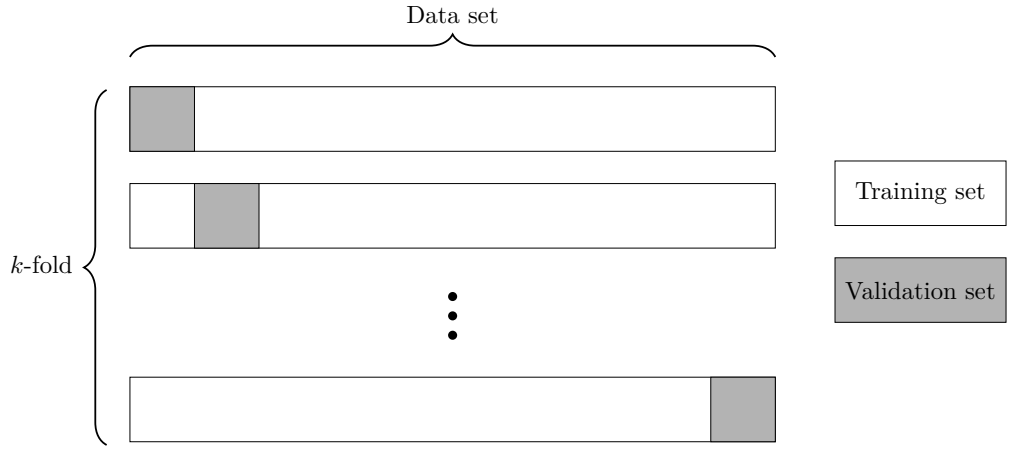


Figure 6.1: Illustration of a k-fold cross validation where the validation set in each fold is made up of a k'th part of the data set.

The CNN models are trained k times; one for each fold. The mean validation accuracy hereof is then computed as:

$$\overline{\mathbb{A}} = \frac{1}{k}\sum_{i=1}^{k}\mathbb{A}_i \tag{6.1}$$

where $\overline{\mathbb{A}}$ is the mean validation accuracy and $\mathbb{A}_i$ is the validation accuracy for the trained CNN model of the i'th fold. The validation accuracy $\mathbb{A}_i$ is computed by evaluating the CNN model on the validation set of the i'th fold. Besides calculating the mean validation accuracy, we also estimated the standard deviation, $\hat{\sigma}$, to examine the stability of the CNN models:

$$\hat{\sigma} = \sqrt{\frac{1}{k}\sum_{i=1}^{k}\left(\mathbb{A}_i - \overline{\mathbb{A}}\right)^2} \tag{6.2}$$

To ensure that we get the best performing CNN model for each fold according to the validation accuracy, the weights of the network are saved after each improved training epoch. It means that after the first epoch the weights of the network are saved, and every time a following epoch gets a greater performance than the previous best one, the weights are overwritten. The improvement check for saving the weights of the best performing network is done by using the performance check in the early stopping procedure, which also is used for stopping the training of the network after a few epochs without improvements. Early stopping checks for improvement in the current validation accuracy, $\mathbb{A}_i$, by comparing it to the best previous epoch, $\mathbb{A}_j$, subtracted a small value, $\varepsilon \geq 0$:

$$\mathbb{A}_i < \max_j \mathbb{A}_j - \varepsilon \qquad j \in \{0, 1, \ldots, i-1\} \tag{6.3}$$

If the validation accuracy of the new epoch is smaller than this, a counter is incremented and when the counter gets over a predefined threshold, the training of the

network is stopped. When the validation accuracy of the new epoch is greater, the weights of the network are saved, and the counter is reset to zero. Based on preliminary experiments, we have concluded that a minimum of 20 training epochs and a counter limit on two where sufficient to ensure a small training loss before stopping. Thus, the training can only be stopped after the 20'th epoch. However, there is no guarantee that the best saved network weights are obtained after the 20'th epoch.

6.2 Room Volume Classification

In this section, the results of the room volume classification for different experiments and Data Set Collections (DSCs) will be presented. Given that the CNN models are trained using supervised learning (See Section 1.3.3), we need to define the classes. The different room volume intervals according to number of classes will be presented in the following.

We have chosen to base the different room volume intervals on the distribution of the room volumes for the 10 140 different rooms in the DSCs with varying room dimensions and fixed wall admittance (See Section 3.3). In Figure 6.2, the distribution of the room volumes is shown by a histogram of the calculated room volumes in these DSCs (See Section 3.3).

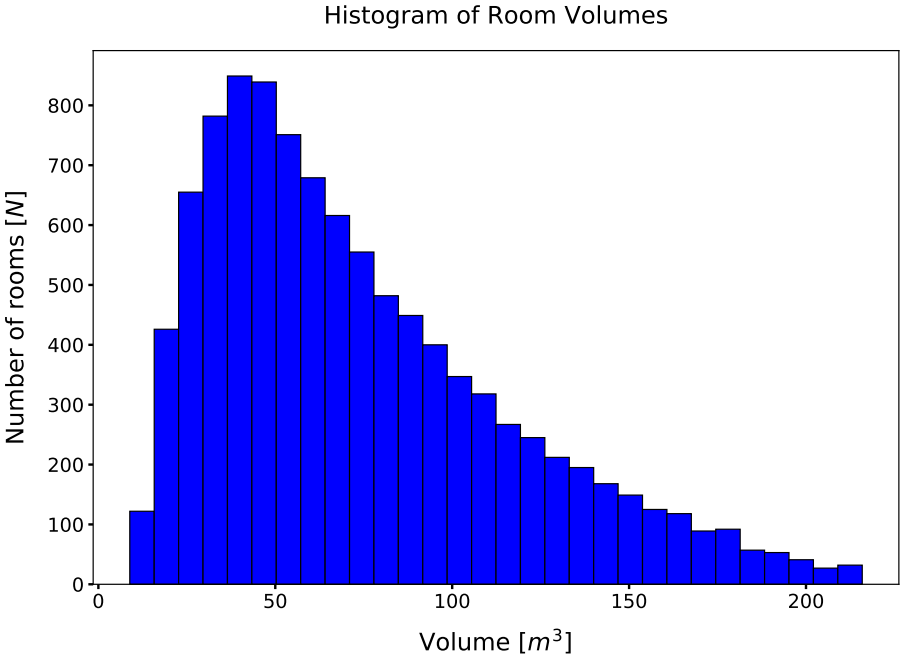


Figure 6.2: Histogram of the room volumes of DSCs with varying room dimensions and fixed wall admittance where there are 10 140 different rooms in each data set.

From Figure 6.2, it seems like the volume of the rooms are distributed as a exponential distribution for the given creation of the room dimensions defined in Section 3.3.4 and the chosen parameters (See Section 3.3.5). We have chosen to make the room volume intervals for the classes such that the classes contain about the same amount of rooms in each class. Given the exponential distribution of the room volumes the size of the intervals varies. This means that the room volume intervals under 90 m^3 consist of smaller intervals with nearly the same number of room volumes and the intervals above 90 m^3 are larger where the room volumes are more scattered. Table 6.1 shows the room volume intervals for the different classes in proportion to the number of classes.

	Number of classes				
	2	3	7	10	20
Room volume intervals $[m^3]$	8-65	8-48	8-32	8-28	8-22
					22-28
			32-44	28-37	28-33
					33-37
			44-57	37-45	37-41
					41-45
		48-85		45-54	45-49
					49-54
			57-73	54-64	54-59
					59-64
	65-220			64-76	64-69
			73-94		69-76
				76-90	76-82
					82-90
		85-220	94-125	90-109	90-99
					99-109
				109-137	109-121
			125-220		121-137
				137-220	137-161
					161-220

Table 6.1: Table of the room volume class intervals for the different number of classes, which are used for the classification of room volumes.

In order to use these class intervals in Table 6.1 for the supervised training and validation we created one-hot vectors (See Definition 4.3), which correspond to the specific class intervals each observation belongs to.

6.2.1 Varying Room Dimensions and Fixed Wall Admittance

This subsection contains the room volume classification results of the three DSCs with varying room dimensions and fixed wall admittance. These three DSCs have different microphone configurations as described in Section 3.3. Each data set in these DSCs consist of one realisation of each room, i.e. each generated room appears only once in each data set.

The classification results of the first DSC with microphones placed in a grid centred in the rooms, referred to as *centred grid*, are given in Table 6.2 (See Section 3.3.1).

	Number of classes				
Mics.	2	3	7	10	20
1	99.05(±0.16)%	98.86(±0.37)%	96.82(±0.37)%	95.50(±0.59)%	88.83(±1.70)%
8	99.18(±0.20)%	98.33(±0.22)%	95.41(±0.42)%	94.02(±1.20)%	87.34(±1.38)%
27	99.48(±0.21)%	98.98(±0.28)%	96.97(±0.51)%	95.40(±0.40)%	89.77(±0.98)%
64	99.40(±0.27)%	98.97(±0.36)%	96.44(±0.55)%	94.91(±0.64)%	89.14(±1.37)%
125	99.47(±0.22)%	99.05(±0.24)%	97.00(±0.47)%	95.30(±0.93)%	90.56(±0.83)%

Table 6.2: The mean validation accuracy and standard deviation for room volume classification of the DSC with varying room dimensions and fixed wall admittance. The microphones are placed in a grid centred in the rooms. There are 10 140 different rooms in each data set. These results are obtained by cross-validation with 10-folds. A batch size of 150 was employed for the training using SGD as the optimisation method. Early stopping is utilised up to a maximum of 50 training epochs.

From Table 6.2, we observe that the number of classes have an impact on the classification accuracy. The classification accuracy decreases in general as the number of classes increases. Furthermore, we observe that the number of classes increases the standard deviation increases as well. For this structured microphone configuration, it seems like the number of microphones only have a very small impact on the accuracy. E.g. the accuracy for 20 classes increases with approximately two percentage points from 1 microphone to 125 microphones. However, the standard deviation for 1 microphone is 1.7 percentage points, and thereby, its best accuracy is similar to the mean validation accuracy for 125 microphones. Also, for the data set with one microphone from this DSC, the microphone is placed such that it has the same distance and orientation to the source for all rooms. Thus, we should be cautionary when comparing with the results of the data set.

The decrement of the accuracy and increment of the standard deviation as the number of classes increases are intuitive since for each number of classes, C, there will be $C-1$ decision boundaries between the C classes. The room volumes in the data sets are dense, and thereby, a lot of observations will be very close to the decision boundaries, which means they will be easily misclassified by the CNN models as their neighbouring classes. For the class intervals under $90\,m^3$ given in Table 6.1, the probability of misclassification is greater compared to the remaining classes due

to the density in the number of observations (See Figure 6.2). In Figure 6.3, the confusion matrix for 20 classes from one of the ten folds and the data set with 27 microphones for this DSC is shown.

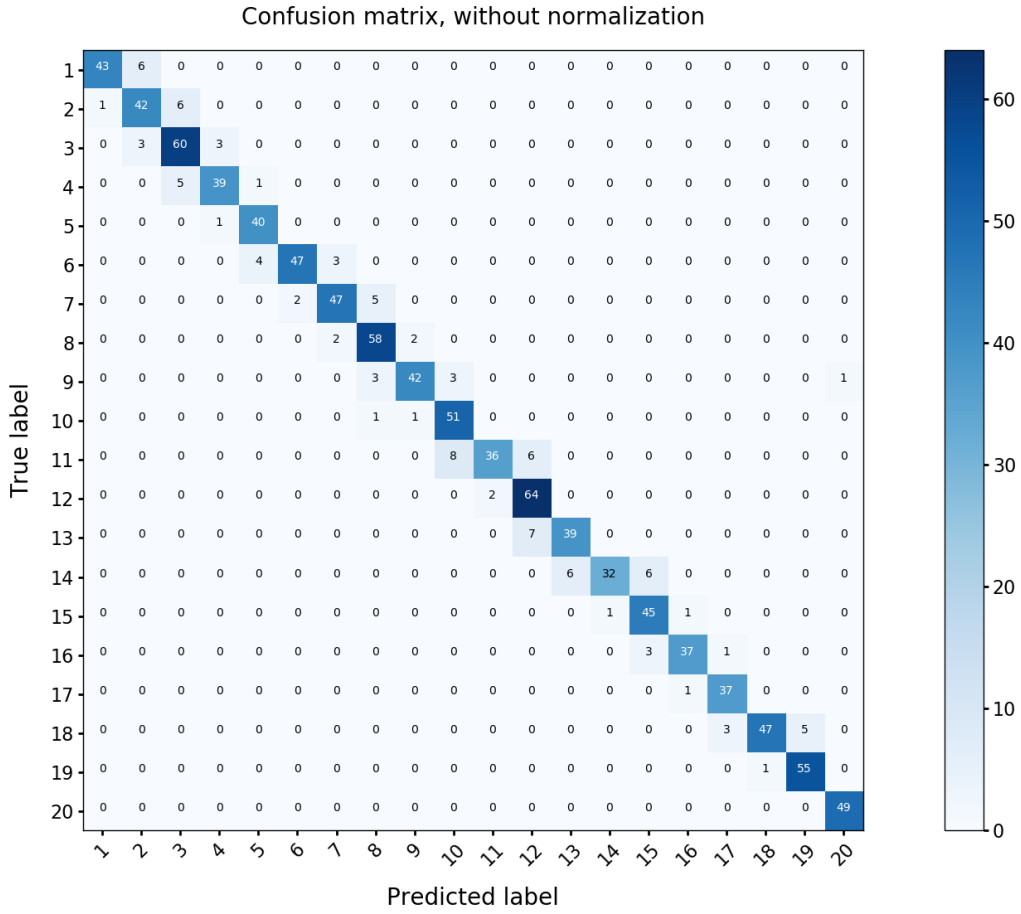


Figure 6.3: Confusion matrix for one of the ten folds room volume classification of 20 classes for the data set with varying room dimensions and fixed wall admittance where 27 microphones placed in a grid centred in the rooms. There are 10 140 different rooms in the data set. The room volume intervals of 20 classes are given in Table 6.1, where class 1 has the room volume interval $0 - 22\ m^3$, etc. There are approximately 51 observations per class on average.

From Figure 6.3, we observe that when a observation is misclassified, it is instead classified as the neighbouring class for all except for two. This is the general picture for the confusion matrices in this experiment, and it suggests that it is the observations close to the decision boundaries that are misclassified as their neighbouring classes by the CNN model.

The classification accuracies for the data sets with 27 microphones from the three DSCs with different microphone configuration are compared to the number of classes

are shown in Figure 6.4. The results for the DSC with microphones placed in a grid centred in one out of eight points in the rooms are referred to as *translated grid* and the DSC with randomly placed microphones in the rooms are referred to as *random subset* (See Section 3.3.2 and 3.3.3 respectively). We have chosen to just show the results for 27 microphones since we observe the same tendencies in classification performance across the number of microphones.

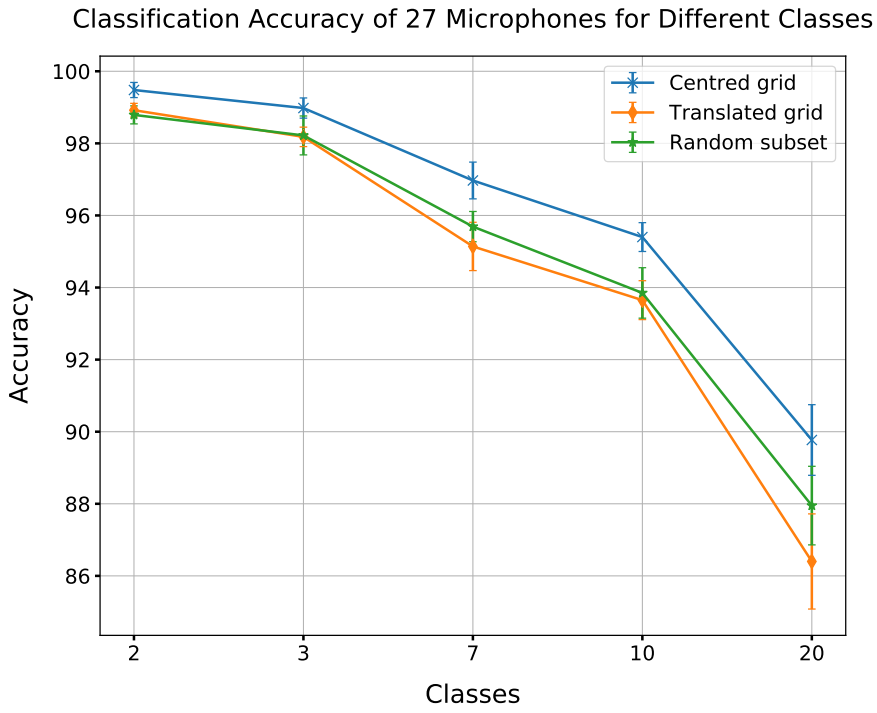


Figure 6.4: Plot of the mean validation accuracy and the standard deviation for the room volume classification against the increasing number of room volume classes. Each curve represents a different data set with varying room dimensions and fixed wall admittance; (**blue**) cubic grid of microphones centred in the room, (**orange**) cubic grid of microphones centred at different positions, and (**green**) randomly placed microphones. All data sets have observations from 27 microphones. There are 10 140 different rooms in each data set.

From Figure 6.4, we observe an identical tendency in the development of the classification accuracies and standard deviations for the three data sets as the number of classes increases. We also observe that the classification of the data set with the centred grids have a bit higher accuracy than the two other data sets independent of the number of classes, which could be caused by some symmetry in the RTFs for this data set given the design of the microphone configuration (See Section 3.3.1). It seems insignificant for the classification of the room volume compared to the number of classes whether the microphones are placed in a translated grids or random subset.

The remaining classification results of the DSC with translated grids and the DSC

with random subsets are given in Table C.2 and Table C.3 respectively. The remaining classification results of these two DSCs have the same tendencies as the classification of the DSC with centred grid, shown in Table 6.2, but with a lower accuracies in general, which also can be seen from Figure 6.4. The classification accuracies for the three DSCs compared to the number of microphones are shown in Figure 6.5 for the 10-class classifications.

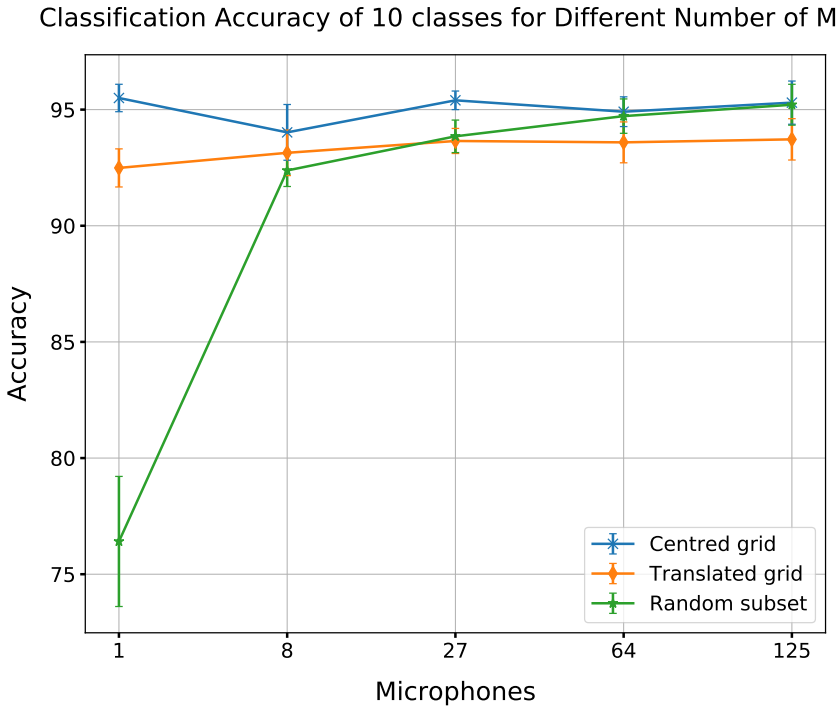


Figure 6.5: Plot of the mean validation accuracy and standard deviation for the room volume classification against the increasing number of microphones. Each curve represents a different data set with varying room dimensions and fixed wall admittance; (**blue**) cubic grid of microphones centred in the room, (**orange**) cubic grid of microphones centred at different positions, and (**green**) randomly placed microphones. A total number of 10 classes, have been used for the room volume classification. There are 10 140 different rooms in each data set.

From Figure 6.5, we observe no improvement in the accuracy or the standard deviation of having more than one microphone except for the DSC with random subset. The accuracy and standard deviation for the DSC with random subsets improves as more microphones are used where the biggest improvement is when we use eight microphones instead of one microphone. It is interesting how the classification accuracy for the data set with 27 microphones in random subsets is equal to the general performance of the DSC with translated grids, and with 64 microphones in random subsets it is equal to the general performance of the DSC with centred grids. Thus, the CNN model becomes independent of the structure in the microphones configuration when the number of microphones increases.

The results of the three DSCs with varying room dimensions and fixed wall admittance have now been presented. In the following subsection, the results of the room volume classification of the DSCs with varying room dimensions and wall admittances will be studied.

6.2.2 Varying Room Dimensions and Wall Admittances

This subsection contains the room volume classification results of the three DSCs with varying room dimensions and wall admittances. These three DSCs have different microphone configurations like for the DSCs with varying room dimensions and fixed wall admittance. Each data set in these DSCs consist of one realisation from each room and wall admittance combination. Given the design of these DSCs described in Subsection 3.3.5, there are 100 realisations of each room where each of these realisations have different wall admittances and vice versa for each wall admittance, i.e. each generated room with a specific wall admittance appears only once in each data set.

The room volume distribution of the rooms in the DSCs is shown in Figure 6.6 to examine how they are distributed compared to the class intervals of the room volumes as defined in Table 6.1.

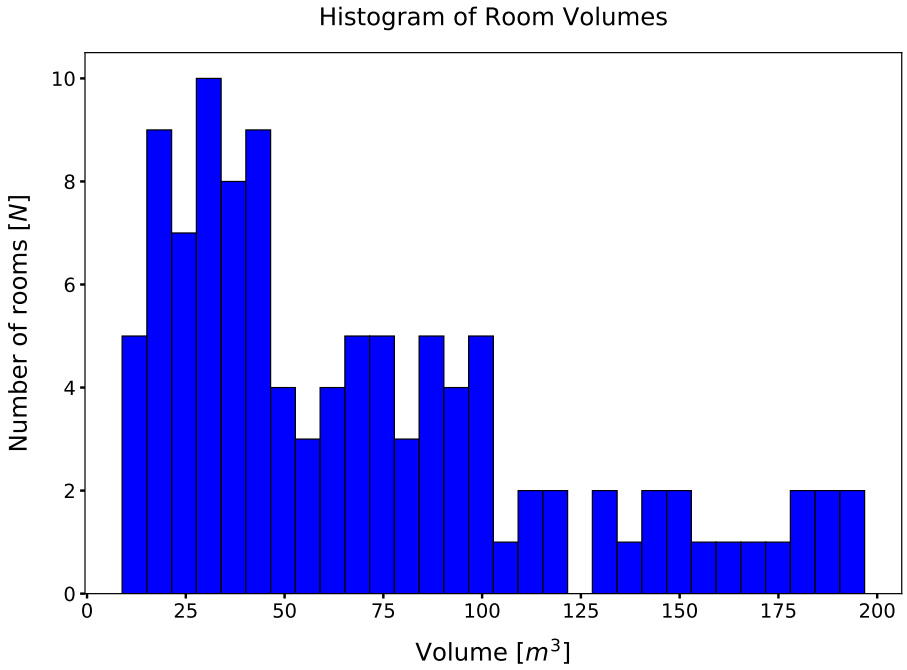
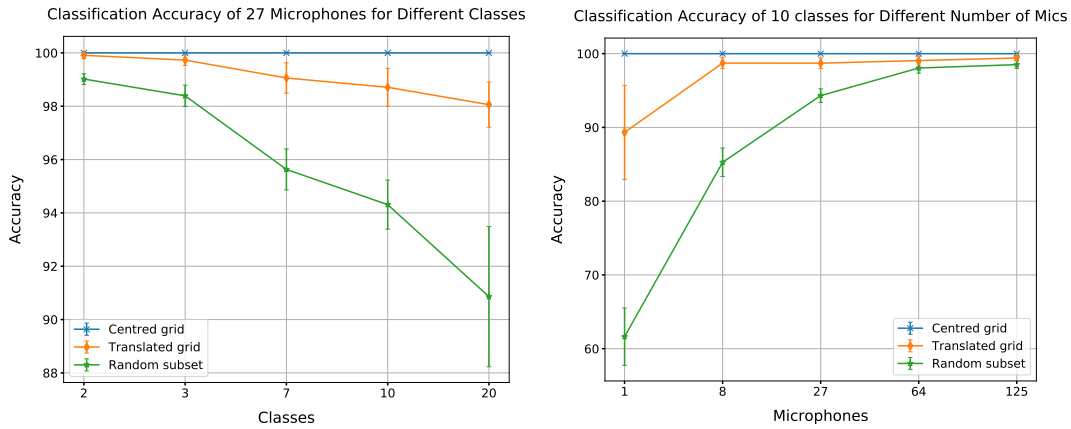


Figure 6.6: Histogram of the rooms volume of data set collections with changing room dimension and wall admittance where there are 108 different rooms in each data set.

As seen in the Figure 6.6, we observe a different distribution of the room volumes of the DSCs compared to the distributions in Figure 6.2. This is because of the different design of the rooms in the data sets where we only have 108 different rooms compared to the 10 140 different rooms for the DSCs with varying room dimensions and fixed wall admittance (See Section 3.3.5), i.e. almost a factor 100 less. Thereby, the DSCs observations will not be as equally divided into the classes in Table 6.1 as for the DSCs with varying room dimensions and fixed wall admittance were.

The comparison of the classification accuracies of the data sets with 27 microphones from the three DSCs and the number of classes are shown in Figure 6.7a, and the classification accuracies according to the number of microphones for the three DSCs are shown in Figure 6.7b for the 10-class classification.



(a) **(b)**

Figure 6.7: Plot of the mean validation accuracy and standard deviation for the room volume classification against (**a**) the increasing number of admittance classes, where all data sets have observations from 27 microphones and (**b**) the increasing number of microphones for the classification of 7 classes of all data sets. Each curve represents a different data set with varying wall admittance and room dimensions; (**blue**) cubic grid of microphones centred in the room, (**orange**) cubic grid of microphones centred at different positions, and (**green**) randomly placed microphones. All data sets have observations from 27 microphones. In each data set, there are 108 different rooms and 100 different wall admittances for each room.

From Figure 6.7a, we observe the accuracy for both the DSC with translated grids and the DSC with random subsets decrease and the standard deviation increases as the number of classes increases, like we saw for all DSCs with varying room dimensions and fixed wall admittance in Figure 6.4. However, the standard deviation increases rapidly for the data set with random subsets from the DSC with varying room dimensions and wall admittances compared to the data set with random subsets from the DSC with varying room dimensions and fixed wall admittance. This suggests that the CNN models struggle learning to distinguish between rooms when the wall admittance varies and there is no structure in the placement of the micro-

phones. When we look at the different DSCs compared to each other, it seems like the microphone configurations have a bigger impact on the classification accuracy than for the DSCs with varying room dimensions and fixed wall admittance shown in Figure 6.4. The accuracy and the standard deviation for the DSC with centred grids in Figure 6.7a is constant and thereby, independent of the number of classes, whereas the accuracy of the DSC with translated grids decrease a little for each increment in the number of classes. Comparing the DSC with random subsets to DSC with translated grid, it seems to decrease twice as fast or more for each increment of the number of classes. Thus, it suggests that having more realisations of each room, even when these have different wall admittances, and structure in placement of the microphones increase performance.

From Figure 6.7b, we observe that the accuracy for both the DSC with translated grids and the DSC with random subsets increase as the number of microphones increases, like we saw for the DSC with varying room dimensions and fixed wall admittance where the microphones are placed in random subsets in Figure 6.5. However, the accuracy for the DSC with translated grids improves insignificantly after eight microphones, and the DSC with random subsets improves little as well after 64 microphones. Based on the observations from Figure 6.7, it seems like the CNN models benefit more from the structure in the microphone placement compared to the DSCs where the wall admittance is kept fixed.

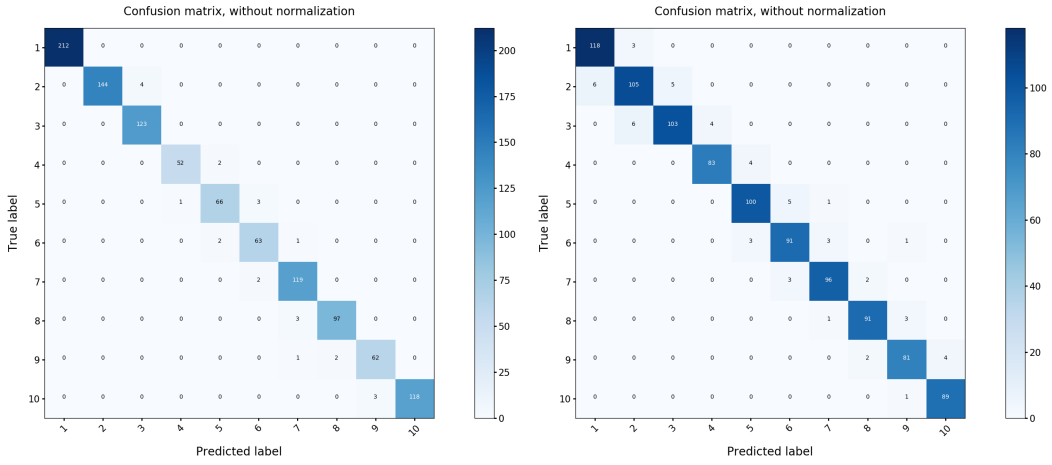
The remaining classification results of the three DSCs are given in Table C.4, C.5, and C.6 for the DSC with centred grids, the DSC with translated grids, and the DSC with random subsets respectively.

We observed in both Figure 6.7a and 6.7b that the classification accuracy of the DSC with centred grids are 100% independent of the number of classes and the number of microphones respectively. When we look at Table C.4, we see that this applies to all the results. Naturally, this should raise some suspicion about whether the CNN model overfits the data. Thus, we have checked if this is the case by first examining the training and validation losses of the model. From this inspection, it did not look like the CNN models were overfitting since the validation loss converge nicely together with the training loss without increasing. Second, we examined the weights of the models to determine what they actually learned from the observations. This suggested that the CNN models learn a filter bank that can "look up" which room the observations are from. In order to confirm this, we set the number of classes equal to the number of different rooms in the DSC, i.e. 108, and the classification hereof still yields 100% accuracy, i.e. each generated room with a specific wall admittance appears only once in each data set.

When examining the filters of first the convolutional layers in the network, it seems like the network learns to remove the damping from the RTFs. We suspected that the

number of filters in the last convolutional layers were too high, since we have more feature maps in the last Convolution Residual (CRes) block than the number of different rooms in the DSC, which might create the filter bank. In order to study this a bit further, we have made a smaller CNN model, where we have reduced the number of filters in the Feature Extraction Part to 8, 16, 32 and 64 for the blocks CRes(1), CRes(2), CRes(3) and CRes(4) respectively and leave out the two first dense layers in Table 5.2. The classification results of this smaller CNN model also yields 100% accuracy for this DSC. Thus, we can conclude that it is possible to distinguish each room from each other. Based upon this, it is still possible that the CNN models are overfitting for the DSC with varying room volume and wall admittance where the microphones are placed in centred grids, since a perfect classification accuracy can still be obtained for an even smaller network architectures. However, considering more realisation of each room might actually be beneficial for the CNN models, and we will return to this shortly.

Confusion matrices for one of the ten folds for 10-class classification are shown in Figure 6.8 of the data set with 27 microphones from the DSC with translated grids in Figure 6.8a and the DSC with random subsets in Figure 6.8b respectively.



(a) Translated grid **(b)** Random subsets

Figure 6.8: Confusion matrices for 10-class room volume classification for one of ten folds of the data sets with 27 microphones from the DSCs with varying room dimensions and wall admittances where the microphones in (**a**) are placed in a grid centre at one of eight points in the rooms and (**b**) are randomly placed in the rooms. The room volume intervals of 20 classes are given in Table 6.1, where class 1 has the room volume interval $0 - 22\ m^3$, etc. There is approximately 108 observations per class on average for each fold. In each data set, there are 108 different rooms and 100 different wall admittances for each room.

From Figure 6.8, we observe that the main part for the misclassified observations are classified as their neighbouring classes, like we observed in the confusion matrix shown in Figure 6.3 for 20 classes and the DSC with centred grids and fixed wall

admittance.

The results described above suggests that having more than one realisation of the rooms in the DSCs improves the accuracy for the structured microphone configurations. However, the results for the DSC with centred grids suggested that increasing the number of realisation for each room seems to make the CNN models overfit the data. This makes sense because the structure and symmetry of the grid results in repeated RTFs, and therefore, the large number of realisations would make the model overfit the data. Hence, 100 observations from each room seems like too many in general. Also, having varying wall admittances seem to prohibit the CNN models from distinguishing between the room volumes for the DSC with random subsets. Thus, we have made a new DSC with a few realisations from each room with a fixed wall admittance for all rooms, which will be presented in the next subsection together with the corresponding classification results.

6.2.3 Multiple Realisations and Additive Complex Gaussian Noise

The tendencies observe in the results of the room volume classification, which were presented in the previous two subsections, suggests that more realisations from each room could improve the classification performance. Therefore, we have chosen to make a new DSC with eight realisations of each room. The chosen data generation set-up is the varying room dimensions and fixed wall admittance where the microphones are placed in a grid, and this microphone grid will then be centred in 8 out of 27 different positions in the rooms in the same way as illustrated in Figure 3.7 for the translated grid. We have chosen the following parameters to make the data sets for the DSC; $N_l = 16$, $N_w = 16$, $N_h = 10$, and $\beta = 0.0002$, such the total number of rooms is 2 560 and the total number of observation is 20 480. The room volume distribution of these rooms is shown in Figure 6.9. The distribution of the room volumes is examined in order to the determine if the chosen class intervals presented in Table 6.1 would give an approximately uniformly distribution of the observations in each class or not.

From Figure 6.9, we observe that the distribution of the room volumes of the DSC are similar to the distribution in Figure 6.2. Thus, an approximately uniform distribution of the observations will be obtained for each class in Table 6.1. The accuracies of the data sets in the DSC according to the number of classes are shown in Figure 6.10.

From Figure 6.10, we observe the same tendencies as in Figure 6.4 and 6.7a where the accuracy decreases as the number of classes increases. We also note the improvement of having more than one microphone were less significant above eight microphones compared to only using one microphone because of the relatively large standard deviation and small accuracy. It suggests that having more than one realisations from each room ensure a higher accuracy in general compared to the DSCs with just one realisation in Figure 6.4, especially for a greater number of classes. Also, the CNN

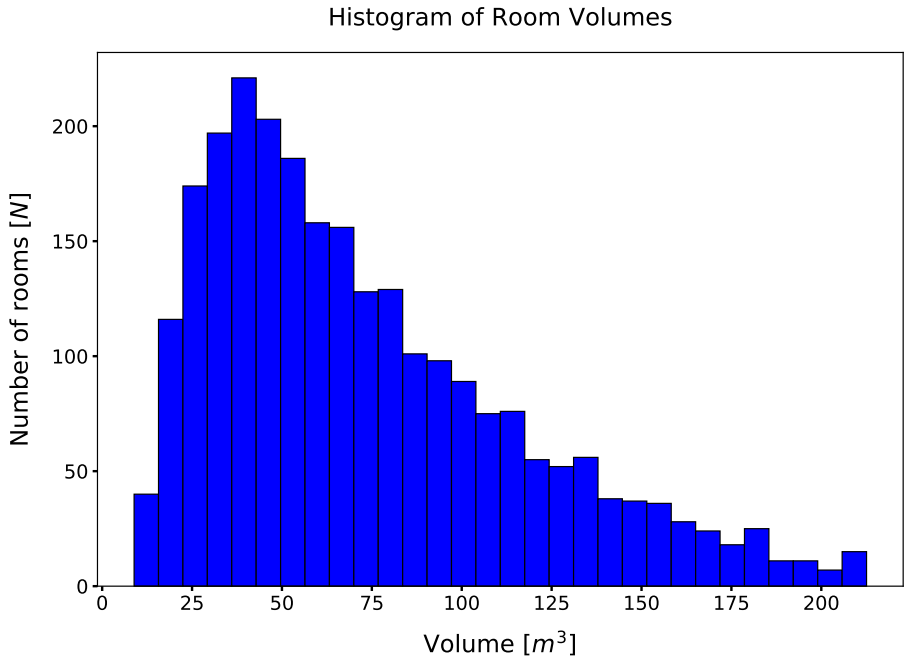


Figure 6.9: Histogram of the rooms volume of DSC with varying room dimension and fixed wall admittance with eight realisations from each room. There are 2 560 different rooms in each data set.

models seem to be less likely to overfit the data compared to the DSCs with varying room dimensions and wall admittances. Hence, it might be the relatively high number of observations that causes the overfitting.

In order to investigate the robustness of the CNN models towards additive noise, we have validated the trained networks on validation sets where complex Gaussian noise has been added in proportion to different SNRs (See Section 5.2.2). The results of the classification hereof on the data set with 27 microphones from the DSC are shown in Figure 6.11 together with the results of the classification of the same set-up but with 15dB SNR complex Gaussian noise added to the training observations as well (See Section 5.2.2).

From Figure 6.11a, we observe that additive complex white noise on the validation set have a big impact on the classification. The smaller SNR the worse the classification gets, where the validation sets with 5dB and 0dB SNR only obtain initial classification accuracies. This corresponds either to classifying all observations as a single class or uniformly random classification, but the general picture from the confusion matrices confirm the former.

From Figure 6.11b, we observe that adding complex Gaussian noise to the training set with 15dB SNR in general increases the classification performance. The classification

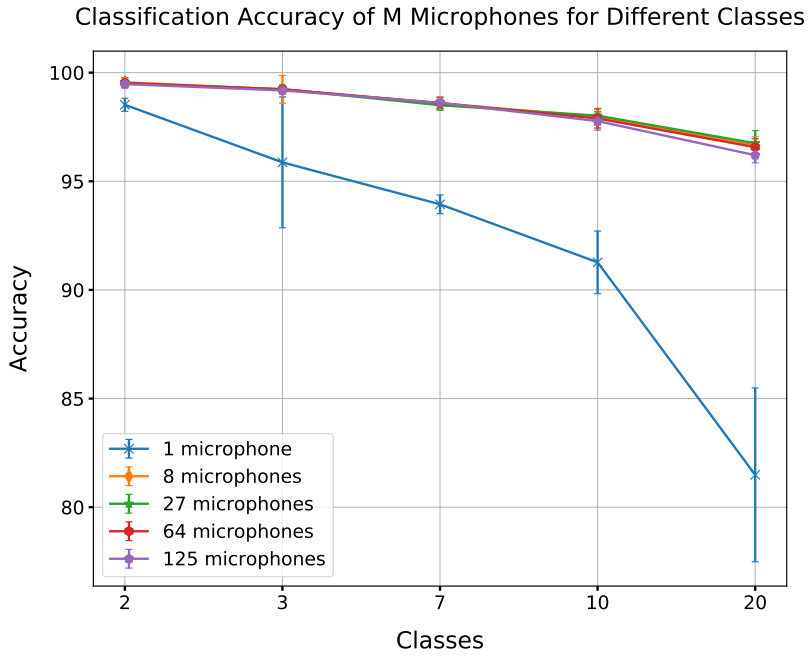
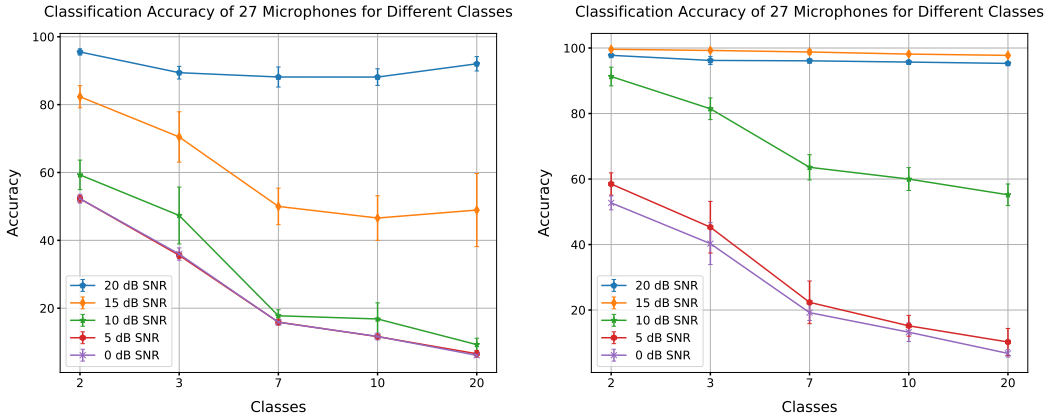


Figure 6.10: Plot of the mean validation accuracy and standard deviation for the room volume classification against the increasing number of room volume classes. The curve represents the data sets from the DSC with varying room dimensions and fixed wall admittance with a cubic grid of microphones centred in 8 out of 27 different positions. There are 2 560 different rooms in each data set.



(a) **(b)**

Figure 6.11: Plot of the mean validation accuracy and standard deviation for the room volume classification against the increasing number of room volume classes. The curve represents the validation sets with different SNRs of added white noise from the data set with varying room dimensions and fixed wall admittance with 27 microphones placed in a grid centred at 8 out of 27 different positions in the rooms. The networks were trained (**a**) on clean simulated RTFs (**b**) on RTFs with added white noise with 15dB SNR. There are 2 560 different rooms in each data set.

accuracies have been improved for especially 15dB and 10dB SNR, where the former corresponds to the same SNR for the training set. However for the validation sets with 20dB SNR, the accuracy is only slightly improved. Thus, it suggests that the CNN models benefit from getting trained with some amount of additive noise, especially for validation sets with similar SNRs since they would reflect similar noise conditions for both training and validation set.

6.3 Wall Admittance Classification

In the previous section, the results of the room volume classification were presented. In this section, the results of the wall admittance, $\mathrm{Re}\{\beta\}$, classification for different simulation experiments and DSCs will be presented. When we mention the wall admittance, we refer to the real component hereof (See Section 3.3.4). Before presenting the results, we will give a description of how the class intervals for the wall admittances are chosen.

As described in Section 3.3, the different wall admittances are made with N_b uniformly spaced values in the interval given in (3.7). Since the values are uniformly spaced, the class intervals are made using the histogram function in the `Python` package `numpy` with the number of bins are equal to the number of classes. This means that the class intervals are of equidistant sizes. Like the class intervals for room volumes, we create one-hot vectors for the classes (See Definition 4.3) in order to use them for supervised training of the CNN models.

6.3.1 Varying Wall admittances and Fixed Room Dimensions

This subsection contains the results for the wall admittance classification of the three DSCs with varying wall admittances and fixed room dimensions where there is 10 000 different wall admittances in each data set. Each of these DSCs are made using a different microphone configuration for the observations (See Section 3.3 for a description hereof). Each data set in these DSCs consist of one realisation of each wall admittance, i.e. each generated wall admittance for the fixed room dimensions appears only once in each data set. The classification results of the DSC with centred grids are given in Table 6.3.

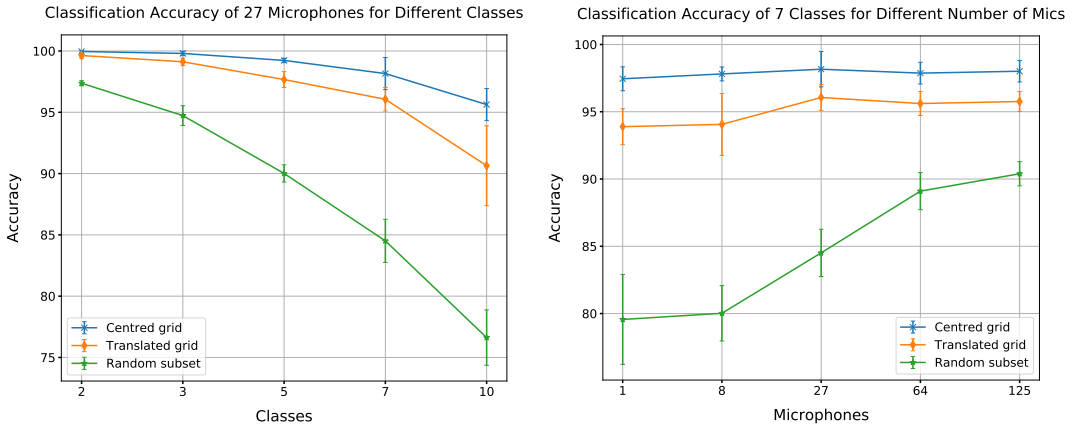
From Table 6.3, we observe that as the number of classes increases the accuracy decreases and the standard deviation increases. For this structured microphone configuration, it seems like the number of microphones only have a very small to no impact on the accuracy. These are similar observations to the ones described in Section 6.2 for room volume classification. This suggest that the number of microphones are insignificant for this microphone configuration. For the data set with one microphone from this DSC, the microphone is placed such that it has the same distance

	Number of classes				
Mics.	2	3	5	7	10
1	99.96(±0.07)%	99.56(±0.27)%	98.91(±0.53)%	97.45(±0.89)%	94.77(±2.56)%
8	99.99(±0.03)%	99.69(±0.19)%	98.98(±0.23)%	97.81(±0.52)%	96.25(±1.34)%
27	99.96(±0.07)%	99.80(±0.18)%	99.23(±0.18)%	98.16(±1.31)%	95.63(±1.31)%
64	99.94(±0.09)%	99.73(±0.12)%	99.18(±0.41)%	97.87(±0.81)%	95.29(±1.54)%
125	99.97(±0.06)%	99.78(±0.13)%	98.95(±0.71)%	98.01(±0.80)%	96.61(±1.40)%

Table 6.3: The mean validation accuracy and standard deviation of the wall admittance classification for the DSC with varying wall admittances and fixed room dimensions, where the microphones are placed in a grid centred in the rooms. There are 10 000 wall admittances in each data set. These results are obtained by cross-validation with 10-folds. A batch size of 150 was employed for the training using SGD as the optimisation method. Early stopping was utilised up to a maximum of 50 training epochs.

and orientation to the source for all rooms. Thus, we should be cautionary when comparing with the results of the data set.

The classification accuracies for the data sets with 27 microphones for the three DSCs compared by the number of classes are shown in Figure 6.12a, and the classification accuracies compared to the number of microphones for the three DSCs are shown in Figure 6.12b for the classification of 10 classes.



(a) **(b)**

Figure 6.12: Plot of the mean validation accuracy and standard deviation for the wall admittance classification against (**a**) the increasing number of admittance classes, where all data sets have observations from 27 microphones and (**b**) the increasing number of microphones for the classification of 7 classes of all data sets. Each curve represents a different data set with varying wall admittance and room dimensions; (**blue**) cubic grid of microphones centred in the room, (**orange**) cubic grid of microphones centred at different positions, and (**green**) random placed microphones. All data sets have observations from 27 microphones. There are 10 000 different rooms in each data set.

From Figure 6.12a, we observe that the accuracy of the DSC with random subsets

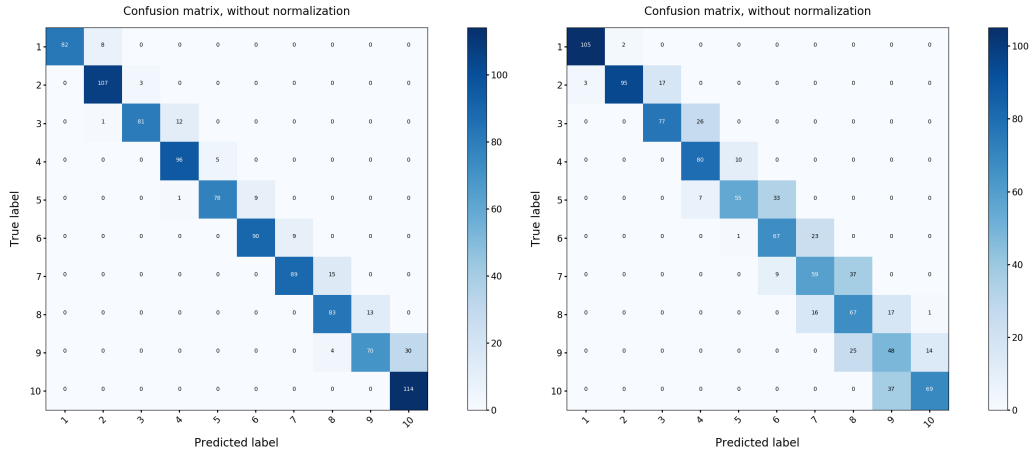
is relative lower than the two other DSCs and decreases faster. The classification accuracy of the DSC with translated grids obtain similar accuracy to the DSC with centred grids but with a bit larger decrement in accuracy as the number of classes increases. The standard deviations of the DSCs increases in general as the number of classes increases. This suggest that the structure in the microphone configurations is more important for the wall admittance classification compared to the room volume classification (See Figure6.4). We believe that the reason for the large decrease in accuracy for the DSC with random subsets is that the CNN model struggles to learn if the resonance frequencies are attenuated by the microphone placement or damping of the walls.

From Figure 6.12b, we observe that the classification accuracy increases little for the two DSCs where the microphones are placed in grids as the number of microphones increases when the standard deviations are taken into account. On the other hand, the accuracy for the DSC with random subsets increases significantly after eight microphones and the standard deviation decreases as well. This suggests that more microphones for classification is beneficial for the CNN model, when the microphones are randomly placed.

The remaining classification results of the DSC with translated grids and the DSC with random subsets are given in Table C.11 and Table C.12 respectively. The remaining classification results of these two DSCs follow the same tendencies in accuracy development regarding the number of classes and the number of microphones as shown in Figure 6.12.

Confusion matrices for one of the ten folds for 10-class classification are shown in Figure 6.13 of the data set with 27 microphones from the DSC with translated grids, which is shown in Figure 6.13a, and the DSC with random subsets, which is shown in Figure 6.13b.

From Figure 6.13, we observe that the main part of misclassified observations are classified as their neighbouring class. Furthermore, we observe a greater amount of misclassified observations for the data set with random subsets than the data set with translated grids. Compared to the general misclassification for room volume classification, we observe similar tendencies for wall admittance classification but with more misclassifications. This could be explained by the relatively narrow interval for the real component of the wall admittance (See Section 3.3.4) since increasing the number of classes make the intervals small and the neighbouring class intervals very similar. Thus, the CNN model struggles to do classification for a greater number of classes, because the problem closer resembles an estimation problem than a classification problem.



(a) Translated grids **(b)** Random subsets

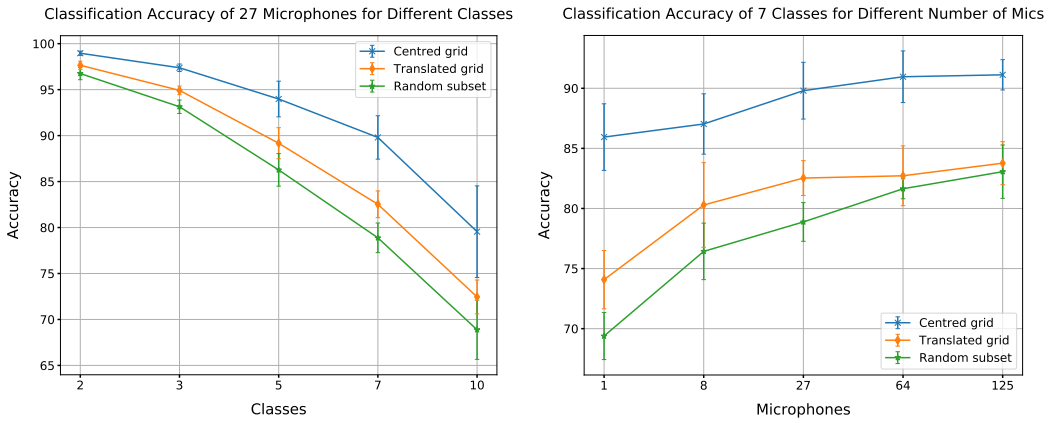
Figure 6.13: Confusion matrices of 10-class wall admittance classification for one of ten folds of the data sets with 27 microphones from the DSCs with varying wall admittances and fixed room dimensions where the microphones in (**a**) are placed in a grid centre at one of eight points in the rooms and (**b**) are randomly placed in the rooms. The beta value intervals of 10 classes are given as follows; class 1 has the beta value interval $1.1 \cdot 10^{-5} - 1.00099 \cdot 10^{-2}$, class 2 has the beta value interval $1.00099 \cdot 10^{-2} - 2.00088 \cdot 10^{-2}$, etc. There are 10 000 different rooms in each data set.

6.3.2 Varying Wall Admittances and Room Dimensions

This subsection contains the results for the wall admittance classification of the three DSCs with varying wall admittances and room dimensions. Each of these DSCs are made using a different microphone configuration for the observations like the DSCs with varying wall admittances and fixed room dimensions in the previous subsection (See Section 3.3). Each data set in these DSCs consist of one realisation for each combination of room and wall admittance. Given the design of these DSCs described in Subsection 3.3.5, there are 108 realisations of each wall admittance where each of these realisations is for different rooms and vice versa for each room, i.e. each generated room with a specific wall admittance appears only once in each data set.

The classification accuracies for the data sets with 27 microphones from the three DSCs compared to the number of classes are shown in Figure 6.14a and the classification accuracies for 7 classes compared to the number of microphones in Figure 6.14b.

In Figure 6.14, we observe that the accuracy decreases and the standard deviation increases as the number of classes increases in Figure 6.14a and the accuracy increases as the number of microphones increases in Figure 6.14b. We also observe that the microphone configuration have a big impact on the classification accuracy for these DSCs. Furthermore, after 64 microphones both the DSC with translated grids and the DSC with random subsets obtain similar accuracies. This suggests that the more structured observations are the better the networks are at classifying



(a) **(b)**

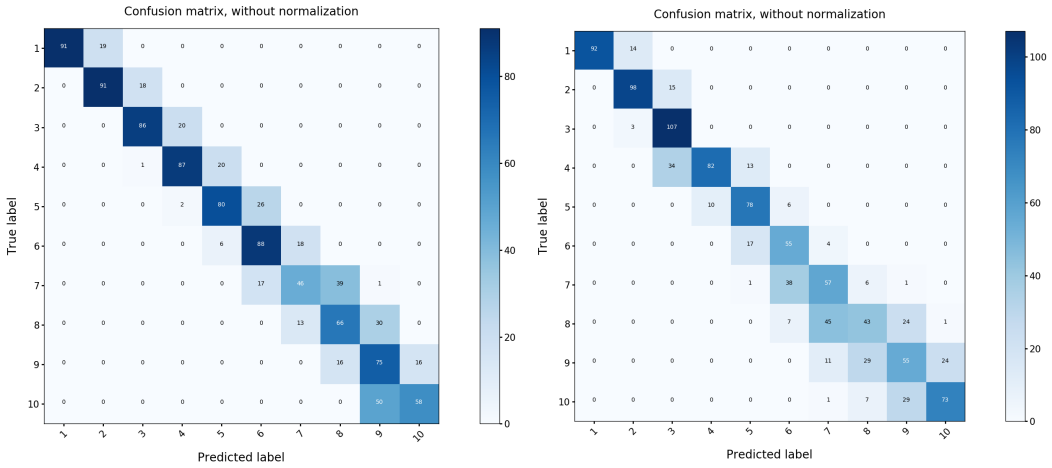
Figure 6.14: Plot of the mean validation accuracy and standard deviation for the wall admittance classification against (**a**) the increasing number of admittance classes, where all data sets have observations from 27 microphones and (**b**) the increasing number of microphones for the classification of 7 classes of all data sets. Each curve represents a different data set with varying wall admittance and room dimensions; (**blue**) cubic grid of microphones centred in the room, (**orange**) cubic grid of microphones centred at different positions, and (**green**) random placed microphones. All data sets have observations from 27 microphones. In each data set, there are 108 different rooms and 100 different wall admittance for each room.

the wall admittance. Unlike for the room volume classification on the same DSCs in Section 6.2.2, we do not observe the same tendencies towards improvement of the accuracies nor signs of overfitting. This suggest that the CNN model can not learn to ignore the affects of the room dimensions on wall admittances, as the CNN model for volume classification ignored the wall admittance.

The remaining classification results of the three DSCs are given in Table C.13, C.14, and C.15 for the DSC with centred grids, the DSC with translated grids and the DSC with random subsets respectively. These remaining classification results of the three DSCs follow the same tendencies of the accuracies development regarding impact of the number of classes and the number of microphones respectively shown in Figure 6.14a and Figure 6.14b.

Confusion matrices for one of the ten folds of 10-class classification are shown in Figure 6.15 of the data set with 27 microphones from the DSC with translated grids, which is shown in Figure 6.15a, and the DSC with random subsets, which is shown in Figure 6.15b.

From Figure 6.15, we observe that the main part of misclassified observations are classified as their neighbouring class. Comparing Figure 6.15 with Figure 6.13, the amount of misclassified observations has increased, and it is especially for the classes with large wall admittances, i.e. the observations are very damped, that the CNN



(a) Microphone grid centred in different points

(b) Random placed microphones

Figure 6.15: Confusion matrices of 10-class wall admittance classification for one of ten folds of the data sets with 27 microphones from the DSCs with varying wall admittances and fixed room dimensions where the microphones in (**a**) are placed in a grid centre at one of eight points in the rooms and (**b**) are randomly placed in the rooms. The beta value intervals of 10 classes are given as follows; class 1 has the beta value interval $1.001 \cdot 10^{-3} - 1.09009 \cdot 10^{-2}$, class 2 has the beta value interval $1.09009 \cdot 10^{-2} - 2.08008 \cdot 10^{-2}$, etc. In each data set, there are 108 different rooms and 100 different wall admittance for each room.

model has difficulties with classifying correctly. Furthermore, we observe that the amount of misclassified observations increases when the microphones are randomly placed instead of being placed in a grid. This suggests that the CNN models are dependent on the room dimensions for classifying the wall admittances, as mentioned above.

6.3.3 Multiple Realisations and Additive White Noise

In an attempt to improve the classification of wall admittances, we have made a data set with more realisations for each room and wall admittance combination. We have chosen to make the new data set such that it has eight realisations for each room and wall admittance combination. We do this to examine if any improvement can be obtained from having more realisations but with fixed room dimensions. The chosen data generation set-up is the varying wall admittances and fixed room dimensions where 27 microphones are placed in a grid and this grid is then centred in 8 out of 27 different positions in the rooms. We have chosen the 27 microphones since it shows the general picture of the classification results and it is comparable to the classification results for room volumes with additive noise. This is done in the same way as illustrated for eight positions in Figure 3.7. We have chosen the number of different wall admittances to be 2 500 such that there is a total number of 20 000 observations in the data set.

In order to investigate the robustness of the CNN models towards additive noise, we have validated the trained networks on validation sets where complex Gaussian noise has been added in proportion to different SNRs (See Section 5.2.2). The results of the classification hereof on the data set are shown in Figure 6.16 together with the results of the classification of the same set-up but with 15dB SNR complex Gaussian noise on the training observations as well (See Section 5.2.2).

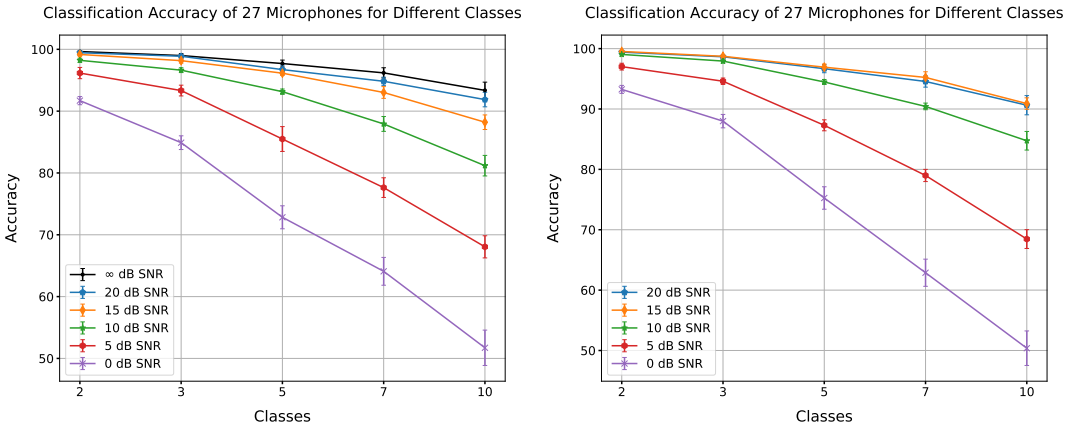


Figure 6.16: Plot of the mean validation accuracy and standard deviation for the wall admittance classification against the increasing number of wall admittance classes. The curve represents the validation sets with different SNRs of added white noise from the data set with varying wall admittances and fixed room dimensions with 27 microphones placed in a grid centred at 8 out of 27 different positions in the rooms. The networks were trained (**a**) on clean simulated RTFs (**b**) on RTFs with added white noise with 15dB SNR. There are 2500 different wall admittances in each data set.

In Figure 6.16a, both the classification accuracies for the validation set with and without additive noise are shown, where the later is referred to as ∞dB SNR in the figure. Compared with the classification accuracy of the DSC with translated grids (See Figure 6.12a), adding more realisations to the data sets seem to have a slight improvement on the accuracy for observations without noise.

From the Figure 6.16a, we observe the same tendencies as for room volume classification in Figure 6.11a, where the accuracy decreases as the SNR decreases. The only difference is that the wall admittance classification performance is relatively better for 5dB and 0dB SNR compared to the same SNR values for room volume classification, where only the initial accuracy were obtained. Thus, the CNN models for wall admittance classification seem to be more robust towards additive noise than the models for room volume. We believe that the CNN models learn to disregard the additive noise since we suspect the models are using the magnitude of the peaks to classify the wall admittances (See Section 1.3.2 and 5.2.2).

As for room volume classification, we observe the same tendencies for wall admittance classification when complex Gaussian noise have have been added to the training set with 15dB SNR. From Figure 6.16b, we observe that the accuracy have been improved with additive Gaussian noise in the training, especially for validation set with similar SNR value. Thus, the CNN models for wall admittance classification seems to benefit from being trained with additive noise, as similarly observed for room volume classification (See Figure 6.11), especially for validation sets with similar SNRs since they would reflect similar noise conditions for both training and validation set.

In this chapter the results of the simulation experiments for both room volume and wall admittance classification have presented. Based upon these results, a general discussion of the report in its entirety will given in the following chapter.

Chapter 7

Discussion

In previous chapters, the proposed CNN models, a description of the simulations experiments, and the results hereof were presented. The results obtained from the simulations experiments will form the basis for the following discussion. First, we give a summary of the main results, followed by a comparison between our contribution and state-of-the-art methods presented in Chapter 1. We will consider the results for the classification of room volume and wall admittance separately in the comparison. The remainder of the chapter will be a formal discussion of the results, assumptions, and applied methods.

The conducted simulation experiments aimed to study the performance of the CNN models for room volume and wall admittance classification (See Chapter 5). Different simulated data set collections (DSCs) were used to examine the impact of the microphone configurations, number of realisations, and simulation set-up on the classification accuracy.

The results of the room volume classifications presented in Section 6.2 showed that an classification accuracy above 90% can be obtained for most number of classes and microphone configurations when considering the DSC with varying room dimensions and fixed wall admittance with a single realisation of each room (See Section 3.3). For a smaller number of classes and more than one microphone, a classification accuracy of roughly 97% is obtained by the CNN models. In general, the misclassified observations of the room volume are primarily classified as the neighbouring classes of the true class independent of the total number of classes (See Section 6.2). This was expected due to the number of rooms in the data sets, the density of the room volumes hereof, and the number of classes since the some of the observation are relatively close to the decision boundaries of the class intervals (See Section 6.2.1). Hence, it followed from the results that the accuracy will decease as the number of classes increases. On the other hand, the number of microphones have a significant impact on the classification accuracy hereof.

The results for the DSCs with varying room dimensions and wall admittances did, however, show great improvements in the classification accuracy compared to the results for the DSCs with varying room dimensions and fixed wall admittance. For the DSC with centred grids, the CNN models obtained perfect classification, i.e. 100% accuracy, independent of the number of classes and microphones. The classification accuracy of the CNN models trained on the two DSCs with respectively translated girds and random subsets were approximately 98% for more than one microphone and a small number of classes. The structure of the microphone configurations showed to have a significant impact on the classification accuracy in general, where the CNN models trained on the DSC with random subsets performed poorly compared to the models trained on the DSC with translated grids.

The DSCs with varying rooms and wall admittance contain only 108 different rooms where there are 100 observations with different wall admittances for each room, in comparison to the 10 140 different rooms in the DSCs with varying room dimensions and fixed wall admittance. The CNN models learn to compensate for the effect of the wall admittances and thereby, only need to learn classifying 108 rooms, which is an approximately factor 100 less compared to the number of rooms for the DSCs with varying room dimensions and fixed wall admittance. However, it was suggested that the relatively high accuracy could be caused by overfitting of the dataset, and therefore, a short analysis hereof were conducted. From examining the weights of the models, it seems like the CNN models learned a filter bank for classifying the rooms instead. A smaller architecture was considered, which still yield a 100% accuracy from the simulations experiments. The analysis did not confirm nor reject the potential overfitting. Hence, further experiments must be conducted in future work. Even though we could not make a final conclusion on the overfitting of the models, the simulation experiments for these DSCs suggested that increasing the number of realisation could be beneficial for the classification.

When a few realisation of each room are added to the data sets with fixed wall admittance. Hence, the number of realisations are increased for both training and validation sets. The classification accuracy is improved especially for 10 and 20 classes, but also fewer different rooms were considered in the DSC hereof. The DSC were generated in a similar way as the DSCs with varying room dimensions and fixed wall admittance, but with approximately four times less rooms and with 8 out of 27 positions for the microphone grids. This way, more realisations of each room and a total number of 20 480 observations in each data set were obtained. The classification accuracy for the DSC with multiple realisation is approximately 99.5% for two classes and decreases to approximately 96% for 20 classes, where the results for one microphone have been omitted. The accuracy for one microphone decreases from 98.5% to 81.5% for 20 classes. Thus, it seems increasing the number of realisations per room improves the room volume classification performance for more than one microphone.

Compared to the results for room volume classification, similar tendencies were observed for the wall admittance classification except with a bit lower accuracy due to a bigger misclassification towards the neighbouring classes. This is especially the case for a relatively large real component of the wall admittance, which results in heavily damped rooms. As mentioned in Section 3.3.4, the higher end of the chosen range of wall admittances violates the assumption of lightly damped rooms, which the model is based upon. This might explain the misclassification of the relatively large wall admittances, but more simulation experiments would have to confirm this. The results of the wall admittance classifications presented in Section 6.3 show that an accuracy above 80% can be obtained for most number of classes and number of microphones for the different DSCs. For the 2-class classification, the accuracy is approximately 99.05% for the DSCs with varying wall admittances and fixed room dimensions and approximately 97.58% for the DSCs with varying wall admittances and room dimensions. Depending on the microphone configuration, the accuracy decrease between 2 and 22 percentage points in general for the DSCs with varying wall admittances and fixed room dimensions and 19 and 28 percentage points in general for the DSCs with varying wall admittances and room dimensions.

As mentioned in Chapter 1, different state-of-the-art approaches have been proposed to solve the problem of classification or estimation of the room acoustic parameters. Most of the papers discussed propose methods for classification or estimation based on reverberant speech signals, instead of the Room Impulse Responses (RIRs) or Room Transfer Functions (RTFs), so they may not seem comparable to our contribution. However, the speech signals used in the papers were all obtained from clean anechoic speech convolved with synthetic or measured RIRs, and not directly measured in the rooms. This means that the reverberant speech could be considered a surjective mapping of the RIRs, a mapping that a DNN can potentially learn. Thus under this assumption, the results from the papers for synthetic RIRs can be compared to our results to some extend, since the RIR is related to the RTF through the Fourier Transform. In this project, the frequency range were limited to 15-300Hz where the papers use a larger frequency range in general. Also, most of the papers consider different feature extractions in the preprocessing of the data and use some or all of them as input to the models where we only consider the log-magnitude response of the RTFs.

The papers by Shabtai et al. [26, 27] trained a Gaussian Mixture Model (GMM) for classification of distinctive rooms from $3\,m^3$ to $41\,606\,m^3$. The first paper by Shabtai et al. [26] trained the GMM on the RIRs directly and only a single RIR per room. For synthetic RIRs, Shabtai et al. [26] obtained a perfect classification of the eight different simulated rooms. A similar results were obtained for the CNN models when trained on the DSC with varying rooms and wall admittances, and the microphones placed in centred grids. For this DSC, we were able to obtained 100% classification

accuracy independent of the number of microphones and classes. Where Shabtei et al. used synthetic RIRs for eight rooms with volumes from $37\,m^3$ to $16\,600\,m^3$, we used synthetic RTFs for 108 room with volumes from $8\,m^3$ to $220\,m^3$. In general, our distribution of room volumes is much narrower and denser compared to the rooms used in any of the mentioned papers (See Chapter 1).

In the second paper by Shabtai et al. [27], where the GMM is trained on reverberant speech, the classification accuracy decreases. When we compare their classification results for synthetic RIRs, it seems that GMM struggles classifying the individual rooms. The classification accuracy does not exceed 50%, while our CNN models obtain an accuracy of approximately 90% or more. Keep in mind, the volume distribution of the rooms is still very different. If Shabtai et al. [27] had considered a narrower volume interval their accuracy might have been higher, or if we had used reverberant speech instead without adapting the CNN models, we might have seen similar decrease in the accuracy.

In parallel to our work, Genovese et al. [8] have proposed a CNN model for blind estimating the room volume based on reverberant speech signals, and considered a similar distribution of the room volumes as Shabtai et al. [27]. The architecture of the CNN model proposed by Genovese et al. [8] and our CNN models have a few design features in common. The CNN model by Genovese et al. [8] produces estimates within approximately a factor of two to the true value of the room volume, which means for smaller rooms the volume are approximately estimated as twice the true value and for larger rooms it is estimate half the true value. Considering only the estimation of the smaller rooms our CNN models for 20 classes would be capable of doing better than a factor two by using the median of each class interval as an estimate of the room volume. E.g. each observation in the class interval $37 - 41\,m^3$ will be estimated as $39\,m^3$, which is the median of this interval, and each observation which are correctly classified to this class gets an error of $0 - 2m^3$. If an observation instead is from the neighbouring class with the interval $33 - 37\,m^3$ and is classified as the aforementioned class it gets an error up to $6\,m^3$. The errors will of course be larger for the class intervals with a bigger range, but it will only be a few observations where the room volume is estimated with an error up to a factor two or more. When examining the confusion matrices of the CNN model by Genovese et al. [8] and the GMM by Shabtai et al. [27], both seems to suffer from large standard deviation, while for our models, it is rather small in comparison. However, this can be due to the use of reverberant speech as training data instead of the RIRs or RTFs directly. Future work would have to confirm this.

Regarding the classification of the wall admittance, it is rather difficult to compare to the performance to any previous proposed method. The method proposed by Santos et al. [25], which estimates the reverberation time using a Long-Short-Term-Memory (LSTM) DNN model, is the most related method as mentioned in Chapter 1. How-

ever, the focus of that paper was to show the improvement using LSTM layers, and how well the model estimated the reverberation time against different rooms and damping coefficients were not studied further. Thus, it would be interesting to do more simulation experiments and try different DNN architectures for classifying or estimating wall admittances in future work.

For all of the aforementioned state-of-the-art approaches above, only the RIRs or speech signals for a single microphone at a time is used. In the simulations experiments conducted, we studied the impact of using more microphones on the classification accuracy. In our study, we found that increasing the number of microphones improved the accuracy in general and especially for a higher number of classes and randomly placed microphones. E.g. the classification accuracy for the DSC with varying room dimensions and fixed wall admittance, and randomly placed microphones showed an improvement of approximately 20 percentage points using 125 microphones instead of just one (See Figure 6.5).

It makes sense from a theoretical perspective that an increment in the number of microphones induces a greater classification accuracy due to the sampling theorem in spatial fields [1, 2]. However, this does not imply that the number of microphones can be increased to infinity. As observed in some figures of Chapter 6, the accuracy tends to converge as the number of microphones increases. This suggest that a sufficient finite number of microphones can be found in order to obtain a desired classification accuracy, which will not necessarily be the same for room volume and wall admittance classification, and microphone configurations.

Increasing the number of microphone especially improved classification of the room volume, since the RTF for each microphone carries information about the eigenmodes. The eigenmodes of a room is determined by the room dimensions only, so from a theoretical perspective the CNN models should be able to perfectly classify the rooms given the RTFs from sufficiently many microphones. Indeed, this is the case if the number of microphones and the placement hereof fulfils the sampling theorem in spatial fields [1, 2]. Hence, the entire sound field can be reproduced up to a certain frequency, and thereby, the whole set of room characteristics can be obtained. It is not only the quantity but also the placement of the microphones that is important since the magnitude of the eigenmodes depend on the source and microphone positions. With this in mind, the CNN models might learn to exploit an undesired symmetry property in the data sets.

In the derivation of the model used for generating our data sets, all six surfaces of the room were assumed to have the same constant wall admittance (See Section 2.1.1). This, combined with the fact that the eigenmodes are a product of cosines, means that there are a few symmetric properties in the simulated rooms, given our choice of generating the rooms (See Section 3.3). E.g. if a simulated room is divided into

eight equally sized cubes, the sound fields of those cubes will be mirror images of each other. This implies that when we generate the DSCs using a microphone grid that is centred in the middle of the room and the source position is kept fixed, we would obtain RTFs that are equal for different microphone positions because of the symmetric properties. E.g. for a grid with 27 centred in the middle of a room will not provide 27 unique RTFs. Thus, we might unintentionally guide the CNN models by feeding the same RTFs for multiple microphone positions or actually prohibit the CNN models from obtaining better classification performance due to the lack of diversity in the data set observations. This symmetric property could explain the relatively classification performance for the DSCs with centred gird in general (See Section 6.2 and 6.3), and especially for the classification of the room volume for the DSC with varying room dimensions and wall admittances (See Table 6.3).

The same symmetric property also occur in the DSCs where the microphone grid is centred at one out of eight points in the rooms. These eight points are made by a second cubic grid in the simulations, where one is picked randomly as the centre for the microphone grid. This means that the microphone grids are placed randomly in one of the eight corners of the room (See Section 3.3.2). Because of the symmetries of the sound field in the room as stated above, moving the grid from corner to corner in a single room will just correspond to shuffling the order of the microphones in the grid. Thus, we might not introduce the amount of randomness in the data set as intended. However, this might not be a big problem, since we only use one realisation for each room and the source are kept fixed in the corner $(0, 0, 0)$ for all rooms. Therefore, two different centred microphone grids from the same room will never occur in the same data set, and if they did, the position of the source relative to the microphone grid will introduce some variety in the RTFs. For the DSCs with eight realisations, the symmetric property do not cause as big a problem since the microphone grids are placed in 8 out of 27 positions in the room. Thus, there is a relatively small probability of placing two microphone grids in e.g. opposite corners.

From the results for both room volume and wall admittances, the microphone configuration with a grid centred in the middle of the rooms gave the the best classification performance, followed by the microphone grids centred in one out of eight points, and the randomly placed microphones, in that order. The accuracy for the randomly placed microphones increased most as the number of microphones increased, while for the other configurations the accuracy remained relatively constant. But considering the symmetric properties discussed above, it might explain the high classification accuracy for the DSCs with microphone grids. The fact that the accuracy is relatively constant for more than one microphone could indicate that the CNN models have learned to exploit these symmetries. The same applies to the DSCs with multiple realisations. So based on the above, more simulation experiments must be conducted in order to examine impact of the symmetric properties on the classification performance in general.

In order to exclude the influence of the symmetric properties on the classification accuracy, future simulation experiments must be conducted. Thus, it would be necessary to recreate the DSCs for those experiments and make sure to simulate the rooms with different wall admittances. Also, it would be preferable to place the source differently, perhaps randomly, in the room. In all of the DSCs, the source have been kept fixed in the same corner of the rooms. First, this is not a very realistic set-up since it is very unlikely a loud speaker would be placed inside the intersection of two walls and the floor. Second, the CNN models will become better at generalising if they are exposed to RTFs for different source position since the CNN models would become independent of the relative positioning of the source and microphone. One can increase the complexity of the simulations by introducing directivity of the source and microphone, simulate furnitures etc. Thus, the next step would be to use real life data sets from different rooms to do better comparison with the already existing methods. However, due the focus on the lower frequency range from 15-300Hz and only considering small rectangular rooms, the number of public available data sets is limited.

As mentioned in Section 6.2.1, the data sets for one microphone in the DSCs with centred grids, the single microphone are placed with the same distance and orientation to source position for all rooms. This position of the microphone was chosen in order to obtain enough information about the odd numbered eigenmodes in the rooms, which would have been missing if the microphone was placed in the middle of the room. However, fixating the relative positioning of the microphone and the source would introduce some structure in the data sets, which could explain the relative high classification performance compared to the DSCs with translated grids and random subsets for one microphones. Generally, a classification accuracy greater than 80% can be obtained for one microphone, when it is not placed at random. This applies to both room volume and wall admittance classification. Hence, it is still possible to do reasonable classification for only a single microphone.

Changing the data set to either new synthetic or measured RTFs implies that the classification of the wall admittance becomes more difficult when all the surfaces no longer are assumed to have the same admittance. One option would be to classify the average wall admittance, but as mentioned further above a general smaller classification accuracy is obtained for wall admittance classification compared to the room volume. Hence, it might be more desirable to consider classification of the reverberation time instead, like Santos and Falk [25], since it is related to both to the room volume and wall admittances.

As a first towards utilising our CNN models on more realistic data sets, we study their robustness against additive white noise (See Section 6.2.3 and 6.3.3). The CNN models were both trained with and without noise, and validated on a validation set

with noise. When adding noise to the training data as well, we expect to improve the classification accuracy for the validation set with the same or close to the same signal-to-noise-ratio (SNR) as the training set. For these simulation experiments, the data sets with 27 microphones from the DSCs with multiple realisations per room are used.

In general, as the SNR decreases so does the accuracy for both room volume and wall admittance classification, when the CNN models are trained without noise. For 10dB SNR the accuracy for room volume classification is below approximately 60% for two class, a minimum drop of approximately 40 percentage points in accuracy, while for wall admittance classification, the drop is only 2 percentage points. It seems like the CNN models for wall admittance classification is more robust against additive complex Gaussian noise in general, when compared to those for room volume classification. From the modal theory presented in Section 2.2, it makes sense why the CNN models for room volume classification are more sensitive to noise. We suspect the CNN models are trying to localise the peaks of the log-magnitude responses of the RTFs (See Section 1.3) since these indicate the eigenmodes of the room from which the room volume can be obtained. The additive noise might confuse the CNN models, which explains the drop in accuracy and why the accuracy increase when the models are trained with noise. For wall admittance classification, we suspect the CNN models are using the magnitude of the eigenmodes, which does not change as much with additive noise (See Figure 5.4), and the noise have a smaller impact on the accuracy. In general, training with additive noise improved the accuracy for both room volume and wall admittance classification compared to training without additive noise and testing with additive noise, but primarily for the validation sets with similar SNR to the training set as expected.

It would be interesting to examine the robustness of the CNN models further when using different preprocessing and data augmentation techniques. E.g. adding noise to only some of the data observation and with different SNR to obtain a bigger diversity in the data. A different approach to both improve the classification accuracy and the robustness of the CNN models would be to use more of the available information in the RTFs. By only using the magnitude response of the RTFs and not the phase response, we only exploit half the amount of data available.

Finally, all of the results presented above and the previous chapter were obtained for a fixed number of training epochs, batch size, learning rate, early stopping, etc. No optimisation of the hyper-parameters for the CNN models were conducted, but the choice of these parameters were based upon results from preliminary simulation experiments. Thus, the obtained results in this report only apply to these specific choices and will most likely change for different batch sizes, learning rates, etc. Optimising the proposed CNN models will be a topic for future work.

Conclusion

A Convolutional Neural Network (CNN) architecture for classification of room characteristics have been proposed. The architecture is inspired by existing designs of high performance Deep Neural Networks in the literature. The feature extraction of the proposed architecture is based on a combination of 1D-Convolutional layers, average- and max-pooling, with residual connections. In order to do classification of the extracted features, a fully connected feed-forward neural network is utilised at the end of the model.

The CNN models are trained using supervised learning to do classification of the room volume and wall admittances based upon the log-magnitude response of Room Transfer Functions (RTFs) observed at different microphone position in a room. As a contribution, a simulation framework have been designed and implemented for generating synthetic RTFs for the Data Set Collections (DSCs), which are used for the training and validation of the CNN models. The model utilised for the simulation is a modal decomposition of the sound field derived from the Helmholtz equation for wave propagation in small rooms. From the validation and verification of the simulation framework, we can conclude that it generates reliable and realistic RTFs for training.

Different simulation experiments were conducted on DSCs with different microphone configurations in order to examine the classification performance of the CNN models. Based on these experiments, we conclude that the classification accuracy of the room volume were at least 90% from the DSCs with one realisations of each room and fixed wall admittance. If we omit the classification of the data sets with only one microphone, the classification accuracy for a smaller number of classes was roughly 97%. The classification accuracy for one microphone is at least 80%, when the microphone is not randomly placed in the rooms. From these results, we can conclude that using more than one microphone yield a higher classification accuracy, and furthermore, it also ensures a smaller accuracy penalty as the number of classes increases.

For classification of the wall admittance, the CNN models achieved an accuracy of at least 80% in general for the DSCs with fixed room dimensions. Similar to the room volume classification, the accuracy decreases for the wall admittance classification as

the number of classes increases. This especially applies to randomly placed microphones with a decrease of 22 percentage points for 10 classes compared to the general classification accuracy of approximately 99% for two classes where the accuracy only decrease with 2 percentage points for the centred microphone grid in the middle of the rooms. For the wall admittance classification, we can conclude that the number of microphones do not have a big impact on the classification accuracy for a smaller number of classes, but its impact increases with the number of classes especially for the randomly placed microphones.

For both room volume and wall admittance classification, it was concluded that the microphone configuration with a grid centred in the middle of the rooms gave the best classification performance, followed by the microphone grids centred in one out of eight points, and the randomly placed microphones, in that order. However, due to different assumptions about the room acoustics, the sound field have symmetric properties, which might be beneficial or disadvantageous for the classification performance. Thus, a final conclusion hereof can not be made based on the current results.

From some of the simulation experiments, generating multiple realisations for each room in the DSCs improves the classification accuracy for both the room volume and wall admittance. Based on these results, the robustness of the CNN models towards additive complex Gaussian noise were examined. The CNN models for room volume classification turned out to be sensitive to additive noise compared to the wall admittance models. It was suggested that the CNN models learned to extract different features depending on the type of classification, and thereby, they are affected differently by additive noise. For an SNR of 10dB on the validation set, the classification accuracy of the room volume dropped approximately 40 percentage points for two classes, while the accuracy for wall admittance classification decreased only by 2 percent point. It was concluded that including a small amount of noise on the training data improved the accuracy for the validation sets with a similar SNR in general.

Summing up, we can conclude that the proposed CNN architecture is capable of classifying the room characteristics, room volume and wall admittance, with a high classification performance, and the misclassification are primarily classified as their neighbouring classes. Generally, the more randomly the microphones were placed the more microphones were necessary to obtain similar accuracy compared to more structured microphone configurations. The CNN models for room volume classification tends to be more sensitive towards noise than the models for wall admittance classification.

Future Work

In Chapter 7, we discussed the results obtained from the simulation experiments (See Chapter 6 for a presentation of the results). In the following, a few ideas for future work will be presented based on the discussion.

Real Life Measurements

First, it would be interesting to study the classification performance of the proposed CNN architectures when exposed to real life measurements of room transfer functions (RTFs). Similar to the simulation experiment with additive noise (See Section 5.2), the first step would be to train the CNN models on simulated RTFs only and then use the real life measured RTFs as a validation set. Afterwards, the CNN models can be trained on the real life measured RTFs in an attempt to improve the classification performance, but this might require some changes to the CNN architectures, e.g. more layers.

Different Parameters and Estimation

In this project, our main focus was on classification of the room volume and the wall admittance. However, different room acoustic parameters related to the absorption in a room exist, e.g. the reverberation time. It would be interesting to study the classification performance for these parameters, and examine if there is any improvement compared to the results presented in this project. As mentioned in the discussion, the wall admittance classification proposed in the report would fail, if the wall admittances are not equal. In that case, classifying the reverberation time instead could solve this problem, since it relates to both the room volume and the absorption, and as we saw for the classification of wall admittances it was dependent on the room dimensions.

Some of the state-of-the-art methods examined in Chapter 1, e.g. Kuster [18], also considered estimation of the room acoustics parameters instead of just classification. In parallel to our work, a recently published paper by Genovese et al. [8] proposed a CNN architecture for estimating the room volume based on reverberant speech signals. Their CNN architecture have few similarities to our design, so it would

be interesting to examine, if the proposed CNN architectures could obtain a better estimation than theirs.

Different Preprocessing and Architectures

Something, we would like to examine further is the impact of the preprocessing on the classification performance. In the simulation experiments, only the log-magnitude response of the RTFs were used, which means we only use half the available data in the RTFs (See Section 1.3.2). Thus, it would be interesting to include the phase response as input to the CNN models and study the classification performance. This will require a redesign of the CNN models as well, in order to include both the magnitude and phase response.

In general, it would be beneficial to study other DNN architectures, such as recurrent neural networks, and try optimise the proposed models in this project. However, this would be a rather comprehensive study and require a lot of preliminary experiments.

Synthesising Room Transfer Functions

As mentioned in Chapter 1, a motivation for studying the capability of the CNN architectures of learning the room acoustic parameters is that the gained knowledge could be used to design a DNN for synthesising Room Transfer Functions (RTFs). Since most methods used in room response equalisation are based on estimates of the RTF or the Room Impulse Response (See Chapter 1), it could beneficial to synthesis RTFs if no information or measurements are available for the listening position.

The main idea is that measurements are available from a few microphones placed arbitrarily in the room where no microphone are placed in the listening position. A DNN model is then trained using these measurement directly or the estimated RTFs for each microphone. Assuming that some geometrical or distance related information can be obtained about the microphone positions and listening position, then the DNN can be conditioned to synthesis the RTF for the listening position.

As a simple proof of concept, we have designed a CNN model based on the complete U-Net architecture [23] with both the down- and up-sampling part. The gated activation-function utilised in WaveNet [28] is used for each up-sampling block in the CNN model in order to introduce the conditioning of the listening position. For this small simulation experiment, the Data Set Collection (DSC) with varying room dimensions, fixed wall admittance and eight realisations for each room was used for the training. A realisation from a room corresponds to a microphone grid centred in one out of 27 positions. Only the data set with a cubic lattice grid and a total of 64 microphones is used.

For each observation in the data set, one of the 64 microphones is picked at random and its RTF is used as the true RTF for the listening position. The remaining 63 microphones are used as input for the network, in order to estimated the RTF of the listing position. Since the microphones are placed in a grid, we create one-hot vectors for the conditioning of the network upon which RTF from a microphone in the grid is missing and should be synthesised (See Chapter 4). The CNN model was trained using the mean square error loss-function, a training set of 15 360 observations, a total number of 50 training epochs, and a batch size of 150 observations. The CNN model was tested on a validation set of 5 120 observations and two examples of the synthesised RTFs against the true RTFs from the validation set are illustrated in Figure 7.1.

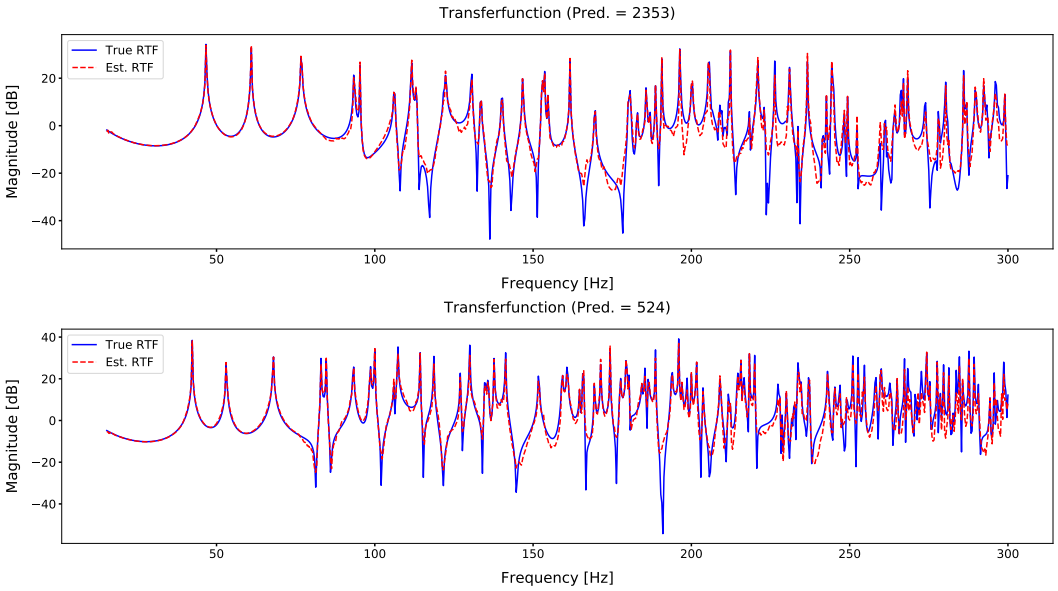


Figure 7.1: Examples of the magnitude response of synthesised RTFs. The red-dashed curve is the synthesised RTF and the blue-solid curve is the true RTF.

The examples in Figure 7.1 show the general picture of the synthesised RTFs from this simulation experiment. Hence, designing a CNN model for synthesising RTFs in future work seems promising and different research topics could be brought up in the field. For instance, it would be interesting to make the CNN model generalise better, by using random placed microphones or by conditioning with e.g. relative distances instead of the one-hot vectors.

Bibliography

[1] T. Ajdler, L. Sbaiz, and M. Vetterli. The Plenacoustic Function and Its Sampling. *IEEE Transactions on Signal Processing*, 54(10):3790–3804, oct 2006. doi: 10.1109/tsp.2006.879280.

[2] T. Ajdler, L. Sbaiz, and M. Vetterli. The Plenacoustic Function and its Sampling. Technical report, Audiovisual Communications Laboratory, EPFL, Lausanne Switzerland, 2006. URL https://infoscience.epfl.ch/record/52075/files/techrep2.pdf.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer New York, 2016. ISBN 1493938436.

[4] Stefania Cecchi, Alberto Carini, and Sascha Spors. Room Response Equalization—A Review. *Applied Sciences*, 8(1), dec 2017. doi: 10.3390/app8010016.

[5] Alan K. Mackworth David L. Poole. *Artificial Intelligence*. Cambridge University Pr., 2017. ISBN 110719539X.

[6] EBU. Tech. 3276 - Listening conditions for the assessment of sound programme material: monophonic and two-channel stereophonic. Technical Document 2nd, European Broadcasting Union, 1998. URL https://tech.ebu.ch/docs/tech/tech3276.pdf.

[7] Gerald B. Folland. *Fourier Analysis and Its Applications*. Orient Black Swan, 2010. ISBN 9780821852088.

[8] Andrea F. Genovese, Hannes Gamper, Ville Pulkki, Nikunj Raghuvanshi, and Ivan J. Tashev. Blind room volume estimation from single-channel noisy speech. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, may 2019. doi: 10.1109/icassp.2019.8682951.

[9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *arXiv e-prints*, art. arXiv:1406.2661, Jun 2014.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv e-prints*, art. arXiv:1512.03385, Dec 2015.

[12] K. Uno Ingard and Philip M. Morse. *Theoretical Acoustics*. PRINCETON UNIV PR, 1987. ISBN 0691024014.

[13] F. Jacobsen and P. M. Juhl. *Fundamentals of General Linear Acoustics*. Wiley, 2013. ISBN 9781118636176.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, art. arXiv:1412.6980, Dec 2014.

[15] Lawrence E. Kinsler, Austin R. Frey, and Alan B. Coppens. *Fundamentals of Acoustics*. PAPERBACKSHOP UK IMPORT, 1999. ISBN 0471847895.

[16] Hans Konrad Knörr. Introduction to partial differential equations, November 2018.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6): 84–90, may 2017. doi: 10.1145/3065386.

[18] Martin Kuster. Reliability of estimating the room volume from a single room impulse response. *The Journal of the Acoustical Society of America*, 124(2): 982–993, aug 2008. doi: 10.1121/1.2940585.

[19] Heinrich Kuttruff. *Room Acoustics: Fourth Edition*. CRC Press, 2007. ISBN 0419245804.

[20] Philip M. Morse and K. Uno Ingard. *Theoretical Acoustics*. Princeton University Press, 1987. ISBN 0691024014.

[21] Ivars Namatēvs. Deep Convolutional Neural Networks: Structure, Feature Extraction and Training. *Information Technology and Management Science*, 20(1), jan 2017. doi: 10.1515/itms-2017-0007.

[22] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing: Pearson New International Edition*. Pearson Education Limited, 2013. ISBN 9781292038155.

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv e-prints*, art. arXiv:1505.04597, May 2015.

[24] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. ImageNet Large Scale Visual Recognition Challenge. *arXiv e-prints*, art. arXiv:1409.0575, Sep 2014.

[25] João F. Santos and Tiago H. Falk. Blind Room Acoustics Characterization Using Recurrent Neural Networks and Modulation Spectrum Dynamics. In *Audio Engineering Society Conference: 60th International Conference: DREAMS (Dereverberation and Reverberation of Audio, Music, and Speech)*, Jan 2016. URL http://www.aes.org/e-lib/browse.cfm?elib=18074.

[26] Noam R. Shabtai, Yaniv Zigel, and Boaz Rafaely. Room volume classification from room impulse response using statistical pattern recognition and feature selection. *The Journal of the Acoustical Society of America*, 128(3):1155, 2010. doi: 10.1121/1.3467765.

[27] Noam R. Shabtai, Yaniv Zigel, and Boaz Rafaely. Towards Room-Volume Classification from Reverberant Speech using Room-Volume Feature Extraction and Room-Acoustics Parameters. *Acta Acustica united with Acustica*, 99(4):658–669, jul 2013. doi: 10.3813/aaa.918644.

[28] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. *arXiv e-prints*, art. arXiv:1609.03499, Sep 2016.

[29] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel Recurrent Neural Networks. *arXiv e-prints*, art. arXiv:1601.06759, Jan 2016.

[30] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional Image Generation with PixelCNN Decoders. *arXiv e-prints*, art. arXiv:1606.05328, Jun 2016.

[31] E. G. Williams. *Fourier Acoustics: Sound Radiation and Nearfield Acoustical Holography*. Elsevier Science, 1999. ISBN 0127539603.

Appendix A

List Overview of Scripts

A zip folder with the code used for this project is attached the to report. The zip folder contains five folders and a `README` file, which describe the folders and files in the zip folder. The five folders are "Experiments", "MasterModule", "Plots", "Scripts", and "OptimiserChangeData". The Experiments-folder contains the four simulation experiment scripts that are use in the project; two for room volume experiments and two for wall admittance experiments. These experiment scripts are listed and described as the first four scripts in Table A.1. The MasterModule-folder contains five different modules with Python classes and functions, which are used in the different scripts. The modules are listed and described in Table A.2. The Plots-folder is for the output of the different scripts. The Script-folder contains the scripts for generating Data Set Collections (DSCs) and making the different plots presented in the report. These script are listed and described as the remaining scripts in Table A.1. The OptimiserChangeData-folder contains the results from the simulation experiment studying the affects of using different optimisation methods. The results are illustrated in Figure 5.3 and discussed in Section 5.1.

In order to run these scripts in Python 3.6, it is necessary to have install the following Python packages: `sklearn` and `Tensorflow API r1.12`. When creating DSC with the generation scripts, the script start by asking questions for specifying simulation set-up and microphone configuration. Based on these the DSC is created and named in the following way: Training_*`set-up`*_cf_*`configuration`*.hdf5. The *`set-up`* types are `room`, `beta` and `mix` for the respectively generation set-ups; varying room dimensions and fixed wall admittance, varying wall admittances and fixed room dimensions, and varying room dimensions and wall admittances. The different *`configuration`*s for the microphone configuration are as follows; `cen_grid` for the configuration where the microphone are placed in a grid centred in the middle of the rooms, `r{a}_move_grid_{b}` for the configuration where the microphones are placed in a grid centred in $\{a\}$ out of $\{b\}$ different positions in the rooms, where $\{a\}$ represent the number of realisations in each room and $\{b\}$ the number of different points the grid can be placed in, and `random` for the configuration where the micro-

phones are placed randomly in the rooms. For running a experiment with a given DSC you write "`python script.py DSC_name`" in the terminal. If you want to run the experiment with noise and want the network to be trained on observations with added complex Gaussian noise as described in Section 5.2.2, then just add "True" after the name of the DSC you want to use.

Table A.1: Table of Python scripts.

File name:	Description:
Training_Room_Vol_Classifier_-Cross-Validation.py	Script for the simulation experiments examining the room volume classification performance using k-fold cross-validation.
Training_Room_Vol_Classifier_-Noise_Cross-Validation.py	Script for the simulation experiments examining the room volume classification performance with additive complex Gaussian noise using k-fold cross-validation.
Training_Room_Admittance_Classifier_Cross-Validation.py	Script for the simulation experiments examining the wall admittance classification performance using k-fold cross-validation.
Training_Room_Admittance_Classifier_Noise_Cross-Validation.py	Script for the simulation experiments examining the wall admittance classification performance with additive complex Gaussian noise using k-fold cross-validation.
Generating_grid_data	Script to make the data sets, where the microphones are placed in a grid inside the room.
Generating_random_data	Script to make the data sets, where the microphones are placed random in the room.
Plot_Train_loss_Adam_vs_SGD	Script for making Figure 5.3.
plot_volume_desity	Script for plotting the room volume densities illustrated in Chapter 6.
Modes_Contours_Example	Script for plotting the equal pressure contours of different modes (Illustrated in Figure 2.2).
RTF_examples_report_plots	Script for creating various different figures through the report. For example Figure 1.2 and Figure 4.2.

Table A.2: Table of Python modules.

File name:	Description:
ExpHelper	Module containing the functions to make and save information and results form the simulation experiments, as well as making DSC files.
Models	Module containing the different CNN models.
PlotTools	Module containing functions for making plots.
SoFIE	Module containing the functions for Sound Fields In Enclosures.
tftools	Module containing the functions made in addition to `TensorFlow`.

Appendix B

Solution Of The Inhomogeneous Helmholtz Equation

In the following appendix, the Green's function of the inhomogeneous Helmholtz equation for rigid walls will be derived. The derivation will be based on a combination of [13, Sec. 8.2] [19, Sec. 3.2]. Let l_x, l_y and l_z denote the room dimensions. Define $\mathcal{B} := (0, l_x) \times (0, l_y) \times (0, l_z)$ as the interior of the room. Then, let $\partial\mathcal{B}$ denote the boundaries of $\mathcal{B}$. Then the Green's function can be obtained from the boundary value problem (BVP):

$$\begin{cases} \Delta\hat{p}(\boldsymbol{r}) + k^2\hat{p}(\boldsymbol{r}) = -\delta(\boldsymbol{r} - \boldsymbol{r}_0) & \text{in } \mathcal{B}, \\ \dfrac{\partial\hat{p}}{\partial n} = 0 & \text{on } \partial\mathcal{B} \end{cases} \tag{B.1}$$

In order to solve this BVP above, we will first solve the homogeneous problem, where the PDE is replaced by the homogeneous Helmholtz equation given by (2.2):

$$\Delta\hat{p}(\boldsymbol{r}) + k^2\hat{p}(\boldsymbol{r}) = 0 \tag{B.2}$$

The homogeneous solution is found by separation of variables, i.e. assume that $\hat{p}(\boldsymbol{r}) = \hat{p}_x(x)\hat{p}_y(y)\hat{p}_z(z)$. Hence, the homogeneous Helmholtz equation can be rewritten as three ordinary differential equations (ODEs):

$$\frac{\partial^2}{\partial x^2}\hat{p}_x(x) + k_x^2\hat{p}_x(x) = 0 \tag{B.3a}$$

$$\frac{\partial^2}{\partial y^2}\hat{p}_y(y) + k_y^2\hat{p}_y(y) = 0 \tag{B.3b}$$

$$\frac{\partial^2}{\partial z^2}\hat{p}_z(z) + k_z^2\hat{p}_z(z) = 0 \tag{B.3c}$$

where $k_x^2 + k_y^2 + k_z^2 = k^2$ must be satisfied. Now, in order to solve the homogeneous problem, one have to solve the system of ODEs above using the same boundary conditions. Since the procedure for solving these ODEs are the same, we will only

derive the solution for (B.3a). The others can be solved by repeating the same steps. The solution formula for an second order ODE, as given in (B.3a), is [13, Sec. 8.2]:

$$\hat{p}_x(x) = Ae^{jk_x x} + Be^{-jk_x x} \tag{B.4}$$

By applying the boundary condition $\frac{\partial}{\partial x}\hat{p}_x(x)\,|_{x=0} = 0$ yields:

$$\begin{aligned}\left.\frac{\partial}{\partial x}\hat{p}_x(x)\right|_{x=0} &= jk_xA - jk_xB = 0\\ &\Rightarrow A = B\end{aligned}$$

Thus, $\hat{p}_x(x) = 2A\cos(k_x x)$. Using the boundary condition, $\frac{\partial}{\partial x}\hat{p}_x(x)\,|_{x=l_x} = 0$, we get the following.

$$\begin{aligned}\left.\frac{\partial}{\partial x}\hat{p}_x(x)\right|_{x=0} &= -2Ak_x\sin(k_xl_x) = 0\\ &\Rightarrow k_xl_x = n_x\pi, \quad n_x \in \mathbb{Z}\end{aligned}$$

Thus, the solution to (B.3a) is given as [13, Sec. 8.2] [19, Sec. 3.2]:

$$\hat{p}_x(x) = \sum_{n_x=0}^{\infty} 2A_{n_x}\cos\left(\frac{n_x\pi}{l_x}x\right) \tag{B.5}$$

where we have restricted n_x to non-negative integers, due to cosine being an even function. Combining the solutions for (B.3) to obtain the solution of (B.1), we get:

$$\hat{p}(\boldsymbol{r}) = \sum_{|N|=0}^{\infty} \Lambda_N\psi_N(\boldsymbol{r}), \quad \text{where} \tag{B.6}$$

$$N := (n_x, n_y, n_z) \in \mathbb{N}_0^3, \quad |N| = n_x + n_y + n_y, \tag{B.7}$$

$$\psi_N(\boldsymbol{r}) = \sqrt{\varepsilon_{n_x}\varepsilon_{n_y}\varepsilon_{n_z}}\cos\left(\frac{n_x\pi}{l_x}x\right)\cos\left(\frac{n_y\pi}{l_y}y\right)\cos\left(\frac{n_z\pi}{l_z}z\right),$$

$$\varepsilon_m = \begin{cases}1 & m = 0\\ 2 & m = 1, 2, \ldots\end{cases}$$

where $\{\psi_N(\boldsymbol{r})\}_N$ is an orthogonal basis in $L^2(\mathcal{B})$ and are called the modes [13, Sec. 8.2] [19, Sec. 3.2]. Thus, the solution can be stated as a linear combination of this basis, i.e. a multivariate Fourier series. Since the modes are orthogonal, we have:

$$\int_{\mathcal{B}} \psi_m\psi_n \,\mathrm{d}V = \begin{cases}V = l_xl_yl_z & \text{for } m = n\\ 0 & \text{for } m \neq n\end{cases} \tag{B.8}$$

Another important property of the modes is that they are each a solution of the PDE (B.2) [13, Sec. 8.2] [19, Sec. 3.2]:

$$\Delta\psi_N + k_N^2\psi_N = 0 \quad \forall N \in \mathbb{N}_0^3 \tag{B.9}$$

We will use these results to obtain a Green's function for the BVP (B.1). First, we want to express the Green's function and the source distribution in terms of the modes, $\{\psi_N(\boldsymbol{r})\}_N$:

$$G_\omega(\boldsymbol{r}, \boldsymbol{r}_0) = \sum_{|N|=0}^{\infty} A_N \psi_N(\boldsymbol{r}) \tag{B.10a}$$

$$-\delta(\boldsymbol{r} - \boldsymbol{r}_0) = \sum_{|N|=0}^{\infty} B_N \psi_N(\boldsymbol{r}) \tag{B.10b}$$

where the equality is in sense of distribution. Multiply (B.10b) by $\psi_M(\boldsymbol{r})$ and integrate on both sides yield:

$$-\psi_M(\boldsymbol{r}_0) = \int_{\mathcal{B}} -\delta(\boldsymbol{r} - \boldsymbol{r}_0)\,\psi_M(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} = \int_{\mathcal{B}} \sum_{|N|=0}^{\infty} B_N \psi_N(\boldsymbol{r})\psi_M(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} = B_M V \tag{B.11}$$

where the last equality follows from (B.8). Hence, we obtain the following:

$$-\delta(\boldsymbol{r} - \boldsymbol{r}_0) = \sum_{|N|=0}^{\infty} -\frac{\psi_N(\boldsymbol{r}_0)}{V}\psi_N(\boldsymbol{r})$$

Now by using (B.10a), (B.9) and (B.2) to obtain:

$$\begin{aligned}
\left(\Delta + k^2\right) G_\omega(\boldsymbol{r}, \boldsymbol{r}_0) &= \left(\Delta + k^2\right) \sum_{|N|=0}^{\infty} A_N \psi_N(\boldsymbol{r}) \\
&= \sum_{|N|=0}^{\infty} A_N \left(\Delta + k_N^2 - k_N^2 + k^2\right) \psi_N(\boldsymbol{r}) \\
&= \sum_{|N|=0}^{\infty} A_N \left(k^2 - k_N^2\right) \psi_N(\boldsymbol{r}) \\
&= \sum_{|N|=0}^{\infty} -\frac{\psi_N(\boldsymbol{r}_0)}{V}\psi_N(\boldsymbol{r})
\end{aligned}$$

Hence from the last equality, we get that:

$$A_N = -\frac{\psi_N(\boldsymbol{r}_0)}{V\left(k^2 - k_N^2\right)}$$

Thus, the Green's function for the BVP (B.1) is given by [13, Sec. 8.2] [19, Sec. 3.2]:

$$G_\omega(\boldsymbol{r}, \boldsymbol{r}_0) = -\frac{1}{V} \sum_{|N|=0}^{\infty} \frac{\psi_N(\boldsymbol{r}_0)}{k^2 - k_N^2} \psi_N(\boldsymbol{r}) \tag{B.12}$$

Hence, we have obtained the Green's function as stated in (2.12). For the BVP with non-rigid walls, i.e. $\dfrac{\partial \hat{p}}{\partial n} = -jk\beta\hat{p}$, the Green's function can be obtained in a similar manner but is more comprehensive. We have excluded the derivation here and instead refer the reader to Kuttruff [19, Ch. 3] for more details, and Morse and Ingard [20, Ch. 9] for a thorough derivation for non-rigid walls.

Appendix C

Results tables

In the following appendix, the cross-validation tables for each of the different Data Set Collections (DSCs), which are described in Section 3.3. The tables are the results from the simulation experiments described in Section 5.2, where 10-fold cross-validation were used to study the classification performance of the proposed CNN models (See Section 5.1). A batch size of 150 was employed for the training of the CNN models using Stochastic Gradient Decent (SGD) as the optimisation method. Early stopping was utilised up to a number 50 of training epochs.

Each table consists of the mean validation accuracy and standard deviation for each data set in a given DSC, and different number of classes. A description of the computation of the mean validation accuracy and standard deviation is given in Section 6.1. The data from the tables are used to make the plots in Chapter 6, which illustrated the general classification performance for the DSCs.

C.1 Room Volume Classification

In the following section, the tables for the room volume classification are given. There will be a short introduction and caption to each table.

C.1.1 Varying Room Dimensions and Fixed Wall Admittance

The results for the three DSCs with varying room dimensions and fixed wall admittance will be given in the following. The three DSCs utilise different microphone configurations as described in Section 3.3. Each data set in the DSCs consists of only one realisation for each room, i.e. each generated room appears only once in each data set.

In Table C.1 the classification results for the first DSC are shown. The microphones are placed in a grid centred in the rooms (See Section 3.3.1).

	Number of classes				
Mics.	2	3	7	10	20
1	99.05(±0.16)%	98.86(±0.37)%	96.82(±0.37)%	95.50(±0.59)%	88.83(±1.70)%
8	99.18(±0.20)%	98.33(±0.22)%	95.41(±0.42)%	94.02(±1.20)%	87.34(±1.38)%
27	99.48(±0.21)%	98.98(±0.28)%	96.97(±0.51)%	95.40(±0.40)%	89.77(±0.98)%
64	99.40(±0.27)%	98.97(±0.36)%	96.44(±0.55)%	94.91(±0.64)%	89.14(±1.37)%
125	99.47(±0.22)%	99.05(±0.24)%	97.00(±0.47)%	95.30(±0.93)%	90.56(±0.83)%

Table C.1: The mean validation accuracy and standard deviation for room volume classification of the DSC with varying room dimensions and fixed wall admittance. The microphones are placed in a grid centred in the rooms. There are 10 140 different rooms in each data set. These results are obtained by cross-validation with 10-folds. A batch size of 150 was employed for the training using SGD as the optimisation method. Early stopping is utilised up to a maximum of 50 training epochs.

In Table C.2 the classification results for the second DSC are shown. The microphones are placed in a grid centred at one out of eight different positions in each of the rooms (See Section 3.3.2).

	Number of classes				
Mics.	2	3	7	10	20
1	98.38(±0.39)%	97.68(±0.46)%	94.45(±0.73)%	92.49(±0.82)%	85.17(±1.65)%
8	98.81(±0.29)%	98.16(±0.34)%	95.27(±0.75)%	93.14(±0.98)%	86.78(±1.40)%
27	98.92(±0.19)%	98.18(±0.27)%	95.14(±0.67)%	93.65(±0.54)%	86.40(±1.32)%
64	98.81(±0.24)%	98.19(±0.34)%	95.56(±0.38)%	93.59(±0.88)%	87.14(±0.90)%
125	98.85(±0.47)%	98.14(±0.51)%	95.25(±0.61)%	93.72(±0.89)%	86.67(±1.09)%

Table C.2: The mean validation accuracy and standard deviation for room volume classification of the DSC with varying room dimensions and fixed wall admittance, where the microphones are placed in a grid centred at one out of eight different positions. There are 10 140 different rooms in each data set. These results are obtained by cross-validation with 10-folds. A batch size of 150 was employed for the training using SGD as the optimisation method. Early stopping was utilised up to a maximum of 50 training epochs.

In Table C.3 the classification results for the third DSC are shown. The microphones are placed at random in the rooms (See Section 3.3.3).

	Number of classes				
Mics.	2	3	7	10	20
1	95.58(±0.78)%	91.30(±0.78)%	81.59(±2.97)%	76.41(±2.80)%	62.38(±5.37)%
8	98.17(±0.38)%	97.64(±0.71)%	93.90(±0.85)%	92.38(±0.69)%	84.99(±1.38)%
27	98.79(±0.25)%	98.22(±0.54)%	95.69(±0.42)%	93.85(±0.70)%	87.95(±1.09)%
64	98.90(±0.22)%	98.46(±0.27)%	96.08(±0.60)%	94.72(±0.74)%	89.54(±0.77)%
125	99.02(±0.23)%	98.59(±0.43)%	96.62(±0.55)%	95.21(±0.88)%	89.62(±1.06)%

Table C.3: The mean validation accuracy and standard deviation for room volume classification of the DSC with varying room dimensions and fixed wall admittance, where the microphones are placed at random in the rooms. There are 10 140 different rooms in each data set. These results are obtained by cross-validation with 10-folds. A batch size of 150 was employed for the training using SGD as the optimisation method. Early stopping was utilised up to a maximum of 50 training epochs.

C.1.2 Varying Room Dimensions and Wall Admittances

The classification results for the three DSCs with varying room dimensions and wall admittance will be given in the following. The three DSCs utilise different microphone configurations as described in Section 3.3. Each data set in the DSCs consists of only one realisation for each room and wall admittance combination, i.e. each generated room with a specific wall admittance appears only once in each data set.

In Table C.4 the classification results for the first DSC are shown. The microphones are placed in a grid centred in the rooms.

	Number of classes				
Mics.	2	3	7	10	20
1	100(±0.00)%	100(±0.00)%	100(±0.00)%	100(±0.00)%	100(±0.00)%
8	100(±0.00)%	100(±0.00)%	100(±0.00)%	100(±0.00)%	100(±0.00)%
27	100(±0.00)%	100(±0.00)%	100(±0.00)%	100(±0.00)%	100(±0.00)%
64	100(±0.00)%	100(±0.00)%	100(±0.00)%	100(±0.00)%	100(±0.00)%
125	100(±0.00)%	100(±0.00)%	100(±0.00)%	100(±0.00)%	100(±0.00)%

Table C.4: The mean validation accuracy and standard deviation for room volume classification of the DSC with varying room dimensions and wall admittances, where the microphones are placed in a grid centred in the rooms. There are 108 different rooms and 100 different wall admittances for each room in each data set. These results are obtained by cross-validation with 10-folds. A batch size of 150 was employed for the training using SGD as the optimisation method. Early stopping was utilised up to a maximum of 50 training epochs.

In Table C.5 the classification results for the second DSC are shown. The microphones are placed in a grid centred at one out of eight different positions in each of the rooms.

	Number of classes				
Mics.	2	3	7	10	20
1	98.68(±0.57)%	97.22(±1.82)%	92.7(±3.70)%	89.32(±6.38)%	93.16(±9.64)%
8	99.89(±0.12)%	99.57(±0.22)%	98.95(±0.56)%	98.72(±0.74)%	97.34(±3.21)%
27	99.91(±0.12)%	99.73(±0.20)%	99.06(±0.57)%	98.71(±0.71)%	98.06(±0.85)%
64	99.94(±0.07)%	99.74(±0.14)%	99.10(±0.42)%	99.06(±0.60)%	97.73(±1.71)%
125	99.94(±0.07)%	99.79(±0.12)%	99.25(±0.40)%	99.41(±0.35)%	98.58(±1.92)%

Table C.5: The mean validation accuracy and standard deviation for room volume classification of the DSC with varying room dimensions and wall admittances, where the microphones are placed in a grid centred at one out of eight different positions. There are 108 different rooms and 100 different wall admittances for each room in each data set. These results are obtained by cross-validation with 10-folds. A batch size of 150 was employed for the training using SGD as the optimisation method. Early stopping was utilised up to a maximum of 50 training epochs.

In Table C.6 the classification results for the third DSC are shown. The microphones are placed at random in the rooms.

	Number of classes				
Mics.	2	3	7	10	20
1	95.19(±0.83)%	88.59(±1.74)%	68.27(±2.86)%	61.63(±3.89)%	38.85(±4.07)%
8	97.88(±0.83)%	95.53(±0.87)%	87.66(±1.88)%	85.27(±1.94)%	77.84(±2.27)%
27	99.02(±0.20)%	98.39(±0.40)%	95.63(±0.77)%	94.31(±0.92)%	90.86(±2.63)%
64	99.24(±0.15)%	99.08(±0.24)%	97.64(±0.72)%	98.05(±0.70)%	96.01(±1.38)%
125	99.38(±0.23)%	99.47(±0.17)%	98.62(±0.36)%	98.51(±0.51)%	98.23(±0.70)%

Table C.6: The mean validation accuracy and standard deviation for room volume classification of the DSC with varying room dimensions and wall admittances, where the microphones are placed at random in the rooms. There are 108 different rooms and 100 different wall admittances for each room in each data set. These results are obtained by cross-validation with 10-folds. A batch size of 150 was employed for the training using SGD as the optimisation method. Early stopping was utilised up to a maximum of 50 training epochs.

C.1.3 Multiple Realisations of Each Room

In the previous sections, the results for classification of the room volume with both fixed and varying wall admittances were presented. The different DSCs used for the simulation experiments, only contained a single realisation of each room and wall admittance combination.

The results for room volume classification of the DSC with multiple realisations are presented in Table C.7. The DSC contains data sets with varying room dimensions and fixed wall admittance. The microphones were placed in grid centred at 8 out of 27 different position in the room, in order to obtain multiple realisations from each room.

	Number of classes				
Mics.	2	3	7	10	20
1	98.52(±0.30)%	95.87(±3.01)%	93.94(±0.43)%	91.27(±1.44)%	81.49(±4.00)%
8	99.55(±0.22)%	99.23(±0.64)%	98.59(±0.24)%	97.97(±0.39)%	96.65(±0.40)%
27	99.50(±0.14)%	99.24(±0.14)%	98.50(±0.24)%	98.02(±0.19)%	96.75(±0.58)%
64	99.53(±0.23)%	99.23(±0.20)%	98.60(±0.27)%	97.89(±0.44)%	96.57(±0.39)%
125	99.47(±0.12)%	99.18(±0.15)%	98.61(±0.27)%	97.77(±0.42)%	96.20(±0.35)%

Table C.7: The mean validation accuracy and standard deviation for room volume classification of the DSC with varying room dimensions and fixed wall admittance, where the microphones are placed in a grid centred at 8 out of 27 different positions. There are 2 560 different rooms in each data set. These results are obtained by cross-validation with 10-folds. A batch size of 150 was employed for the training using SGD as the optimisation method. Early stopping was utilised up to maximum of 50 training epochs.

C.1.4 Noise

In order to investigate the robustness of the CNN models towards additive noise, we have validated the trained networks on validation sets where complex white noise have been added in proportion to different SNRs as described in Section 5.2.2. The classification hereof are made on the data set with 27 microphones from the DSC with multiple realisation of each room and the results are given in Table C.8.

	Number of classes				
SNR	2	3	7	10	20
20	95.52(±0.92)%	89.41(±1.87)%	88.14(±2.96)%	88.12(±2.45)%	92.03(±2.12)%
15	82.33(±3.26)%	70.47(±7.42)%	50.00(±5.38)%	46.58(±6.57)%	48.92(±10.77)%
10	59.30(±4.35)%	47.33(±8.37)%	17.77(±1.78)%	16.83(±4.75)%	9.27(±1.89)%
5	52.28(±1.02)%	35.58(±0.89)%	15.90(±0.86)%	11.63(±0.54)%	6.58(±0.98)%
0	52.27(±1.29)%	35.96(±1.81)%	15.86(±0.76)%	11.68(±1.02)%	6.12(±0.45)%

Table C.8: The mean validation accuracy and standard deviation for room volume classification evaluated on validation sets with added noise in proportion to different SNRs for the data set with varying room dimensions and fixed wall admittance where the 27 microphones are placed in a grid centred at 8 out of 27 different positions. There are 2 560 different rooms in the data set. These results are obtained by cross-validation with 10-folds. A batch size of 150 was employed for the training using SGD as the optimisation method. Early stopping was utilised up to a maximum of 50 training epochs.

In Table C.9 the classification results of the same set-up but with 15dB SNR complex white noise on the training observations as well.

	Number of classes				
SNR	2	3	7	10	20
20	97.77(±0.37)%	96.21(±1.25)%	96.09(±0.63)%	95.71(±0.50)%	95.30(±0.58)%
15	99.61(±0.14)%	99.26(±0.22)%	98.79(±0.66)%	98.15(±0.24)%	97.75(±0.45)%
10	91.30(±2.85)%	81.45(±3.31)%	63.58(±3.86)%	59.99(±3.49)%	55.19(±3.30)%
5	58.48(±3.41)%	45.30(±7.88)%	22.35(±6.51)%	15.19(±3.18)%	10.24(±4.14)%
0	52.72(±2.11)%	40.28(±6.40)%	19.21(±2.41)%	13.24(±2.85)%	6.76(±1.12)%

Table C.9: The mean validation accuracy and standard deviation for room volume classification with added noise at 15dB SNR and evaluated on validation sets with added noise in proportion to different SNRs for the data set with varying room dimensions and fixed wall admittance where the 27 microphones are placed in a grid centred at 8 out of 27 different positions. There are 2 560 different rooms in the data set. These results are obtained by cross-validation with 10-folds. A batch size of 150 was employed for the training using SGD as the optimisation method. Early stopping was utilised up to a maximum of 50 training epochs.

C.2 Beta Value Classification

In the following section, the tables for the wall admittance classification are given. There will be a short introduction and caption to each table.

C.2.1 Varying Wall Admittance and Fixed Room Dimensions

The results for the three DSCs with varying wall admittances and fixed room dimensions will be given in the following. The three DSCs utilise different microphone configurations as described in Section 3.3. Each data set in the DSCs consists of only one realisation for each room, i.e. each generated room appears only once in each data set.

In Table C.10 the classification results for the first DSC are shown. The microphones are placed in a grid centred in the rooms.

	Number of classes				
Mics.	2	3	5	7	10
1	99.96(±0.07)%	99.56(±0.27)%	98.91(±0.53)%	97.45(±0.89)%	94.77(±2.56)%
8	99.99(±0.03)%	99.69(±0.19)%	98.98(±0.23)%	97.81(±0.52)%	96.25(±1.34)%
27	99.96(±0.07)%	99.80(±0.18)%	99.23(±0.18)%	98.16(±1.31)%	95.63(±1.31)%
64	99.94(±0.09)%	99.73(±0.12)%	99.18(±0.41)%	97.87(±0.81)%	95.29(±1.54)%
125	99.97(±0.06)%	99.78(±0.13)%	98.95(±0.71)%	98.01(±0.80)%	96.61(±1.40)%

Table C.10: The mean validation accuracy and standard deviation of the wall admittance classification for the DSC with varying wall admittances and fixed room dimensions, where the microphones are placed in a grid centred in the rooms. There are 10 000 wall admittances in each data set. These results are obtained by cross-validation with 10-folds. A batch size of 150 was employed for the training using SGD as the optimisation method. Early stopping was utilised up to a maximum of 50 training epochs.

In Table C.11 the classification results for the second DSC are shown. The microphones are placed in a grid centred at one out of eight different positions in each of the rooms.

	Number of classes				
Mics.	2	3	5	7	10
1	99.60(±0.17)%	98.56(±0.50)%	96.24(±1.27)%	93.89(±1.34)%	86.20(±1.64)%
8	99.74(±0.13)%	99.10(±0.28)%	97.39(±1.03)%	94.06(±2.30)%	88.58(±2.68)%
27	99.63(±0.23)%	99.12(±0.30)%	97.67(±0.65)%	96.06(±0.96)%	90.64(±3.26)%
64	99.73(±0.15)%	99.23(±0.35)%	97.39(±0.51)%	95.61(±0.89)%	90.32(±2.58)%
125	99.79(±0.14)%	99.33(±0.29)%	97.53(±0.85)%	95.76(±0.74)%	91.24(±2.53)%

Table C.11: The mean validation accuracy and standard deviation of the wall admittance classification for the DSC with varying wall admittance and fixed room dimensions, where the microphones are placed in a grid centred at one out of eight different positions. There are 10 000 wall admittances in each data set. These results are obtained by cross-validation with 10-folds. A batch size of 150 was employed for the training using SGD as the optimisation method. Early stopping was utilised up to a maximum of 50 training epochs.

In Table C.12 the classification results for the third DSC are shown. The microphones are placed at random in the rooms.

	Number of classes				
Mics.	2	3	5	7	10
1	96.92(±0.62)%	93.85(±1.22)%	87.03(±1.62)%	79.56(±3.34)%	65.37(±4.28)%
8	96.73(±0.48)%	93.83(±1.13)%	86.98(±1.68)%	80.02(±2.06)%	70.77(±2.67)%
27	97.37(±0.21)%	94.73(±0.81)%	90.01(±0.70)%	84.51(±1.76)%	76.62(±2.26)%
64	98.07(±0.43)%	96.22(±0.43)%	92.29(±0.77)%	89.10(±1.38)%	81.93(±1.97)%
125	98.33(±0.33)%	96.58(±0.53)%	93.62(±0.55)%	90.39(±0.90)%	84.79(±1.92)%

Table C.12: The mean validation accuracy and standard deviation of the wall admittance classification for the DSC with varying wall admittances and fixed room dimensions, where the microphones are placed at random in the rooms. There are 10 000 wall admittances in each data set. These results are obtained by cross-validation with 10-folds. A batch size of 150 was employed for the training using SGD as the optimisation method. Early stopping was utilised up to a maximum of 50 training epochs.

C.2.2 Varying Room Dimensions and Wall Admittances

The classification results for the three DSCs with varying room dimensions and wall admittance will be given in the following. The three DSCs utilise different microphone configurations as described in Section 3.3. Each data set in the DSCs consists of only one realisation for each room and wall admittance combination, i.e. each generated room with a specific wall admittance appears only once in each data set.

In Table C.13 the classification results for the first DSC are shown. The microphones are placed in a grid centred in the rooms.

	Number of classes				
Mics.	2	3	5	7	10
1	98.49(±0.44)%	96.78(±0.66)%	92.93(±0.94)%	85.94(±2.77)%	75.34(±4.10)%
8	98.77(±0.35)%	96.85(±0.71)%	92.57(±1.07)%	87.03(±2.51)%	79.65(±4.66)%
27	98.96(±0.23)%	97.38(±0.41)%	93.98(±1.94)%	89.80(±2.36)%	79.55(±4.98)%
64	99.06(±0.30)%	97.60(±0.46)%	94.41(±0.93)%	90.96(±2.15)%	83.15(±3.36)%
125	99.01(±0.34)%	97.72(±0.42)%	93.86(±0.82)%	91.12(±1.26)%	82.75(±1.92)%

Table C.13: The mean validation accuracy and standard deviation of the wall admittance classification for the DSC with varying wall admittances and room dimensions, where the microphones are placed in a grid centred in the rooms. There are 108 different rooms and 100 different wall admittances for each room in each data set. These results are obtained by cross-validation with 10-folds. A batch size of 150 was employed for the training using SGD as the optimisation method. Early stopping was utilised up to a maximum of 50 training epochs.

In Table C.14 the classification results for the second DSC are shown. The microphones are placed in a grid centred at one out of eight different positions in each of the rooms.

	Number of classes				
Mics.	2	3	5	7	10
1	96.53(±0.49)%	92.78(±0.73)%	84.11(±1.07)%	74.08(±2.43)%	63.41(±2.29)%
8	97.70(±0.43)%	95.28(±0.60)%	88.72(±1.10)%	80.29(±3.52)%	70.89(±4.08)%
27	97.65(±0.44)%	94.92(±0.47)%	89.17(±1.69)%	82.53(±1.45)%	72.45(±1.84)%
64	97.61(±0.39)%	95.43(±0.63)%	89.14(±1.81)%	82.72(±2.49)%	72.91(±2.75)%
125	97.87(±0.47)%	95.48(±0.65)%	90.11(±0.92)%	83.77(±1.79)%	74.01(±1.92)%

Table C.14: The mean validation accuracy and standard deviation of the wall admittance classification for the DSC with varying wall admittance and room dimensions, where the microphones are placed in a grid centred at one out of eight different positions. There are 108 different rooms and 100 different wall admittances for each room in each data set. These results are obtained by cross-validation with 10-folds. A batch size of 150 was employed for the training using SGD as the optimisation method. Early stopping was utilised up to a maximum of 50 training epochs.

In Table C.15 the classification results for the third DSC are shown. The microphones are placed at random in the rooms.

	Number of classes				
Mics.	2	3	5	7	10
1	94.91(±0.77)%	89.13(±0.78)%	78.68(±1.69)%	69.29(±1.96)%	56.38(±2.43)%
8	96.25(±0.39)%	92.58(±0.95)%	83.99(±1.87)%	76.43(±2.35)%	66.34(±2.03)%
27	96.75(±0.67)%	93.14(±0.73)%	86.27(±1.77)%	78.88(±1.61)%	68.88(±3.22)%
64	96.94(±0.71)%	93.77(±0.49)%	88.09(±1.94)%	81.64(±0.83)%	73.79(±3.42)%
125	97.25(±0.39)%	94.55(±0.81)%	89.10(±2.16)%	83.06(±2.22)%	75.81(±2.43)%

Table C.15: The mean validation accuracy and standard deviation of the wall admittance classification for the DSC with varying wall admittances and room dimensions, where the microphones are placed at random in the rooms. There are 108 different rooms and 100 different wall admittances for each room in each data set. These results are obtained by cross-validation with 10-folds. A batch size of 150 was employed for the training using SGD as the optimisation method. Early stopping was utilised up to a maximum of 50 training epochs.

C.2.3 Multiple Realisations of Each Room and Noise

In an attempt to improve the classification, we have made a data set with few realisations for each room and admittance combination. The data set are made for varying wall admittances and fixed room dimensions, where 27 microphones are placed in a grid and this grid is then centred in 8 out of 27 different positions in the rooms. In order to investigate the robustness of the CNN models towards additive noise, we have validated the trained networks on validation sets where complex white noise has been added in proportion to different SNRs as described in Section 5.2.2. The results of the classification hereof are given in Table C.16.

	Number of classes				
SNR	2	3	5	7	10
∞	99.64(±0.12)%	98.99(±0.32)%	97.70(±0.56)%	96.19(±0.83)%	93.36(±1.33)%
20	99.45(±0.14)%	98.88(±0.18)%	96.72(±0.73)%	94.83(±0.66)%	91.87(±1.17)%
15	99.17(±0.23)%	98.17(±0.40)%	96.14(±0.37)%	93.02(±0.95)%	88.21(±1.18)%
10	98.22(±0.33)%	96.63(±0.39)%	93.15(±0.43)%	87.93(±1.20)%	81.18(±1.66)%
5	96.16(±0.91)%	93.33(±0.87)%	85.49(±2.01)%	77.63(±1.59)%	68.05(±1.79)%
0	91.70(±1.02)%	84.90(±1.95)%	72.83(±2.31)%	64.09(±1.08)%	51.72(±3.41)%

Table C.16: The mean validation accuracy and standard deviation of the wall admittance β classification evaluated on validation sets with added noise in proportion to different SNRs for the data set with varying wall admittances and fixed room dimensions where the 27 microphones are placed in a grid centred at 8 of 27 different positions. There are 2 500 different wall admittances in the data set. These results are obtained by cross-validation with 10-folds. A batch size of 150 was employed for the training using SGD as the optimisation method. Early stopping was utilised up to a maximum of 50 training epochs.

In Table C.17 the classification results of the same set-up but with 15dB SNR complex white noise on the training observations as well.

	Number of classes				
SNR	2	3	5	7	10
20	99.42(±0.11)%	98.65(±0.25)%	96.68(±0.67)%	94.56(±0.94)%	90.63(±1.59)%
15	99.53(±0.11)%	98.73(±0.18)%	96.93(±0.52)%	95.23(±0.94)%	90.90(±0.95)%
10	99.04(±0.25)%	97.94(±0.34)%	94.48(±0.41)%	90.43(±0.55)%	84.74(±1.54)%
5	97.02(±0.56)%	94.60(±0.53)%	87.30(±0.91)%	78.99(±1.01)%	68.46(±1.56)%
0	93.25(±0.66)%	87.99(±1.11)%	75.26(±1.86)%	62.88(±2.25)%	50.38(±2.86)%

Table C.17: The mean validation accuracy and standard deviation of the wall admittance β classification train with added noise at 15dB SNR and evaluated on validation sets with added noise in proportion to different SNRs for the data set with varying wall admittances and fixed room dimensions where the 27 microphones are placed in a grid centred at 8 of 27 different positions. There are 2 500 different wall admittances in the data set. These results are obtained by cross-validation with 10-folds. A batch size of 150 was employed for the training using SGD as the optimisation method. Early stopping was utilised up to a maximum of 50 training epochs.