Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

Bror-Lauritz Størkersen

A Feature Extraction Framework for Measuring Auditory Similarity Between Sounds

Master's thesis in Information Security
Supervisor: Lasse Øverlier
June 2022

NTNU
Kunnskap for en bedre verden

Bror-Lauritz Størkersen

A Feature Extraction Framework for Measuring Auditory Similarity Between Sounds

Master's thesis in Information Security
Supervisor: Lasse Øverlier
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

Abstract

Every day, thousands of hours of audio are recorded in the form of raw audio recordings and video. Sometimes, this audio has to be annotated and transcribed to help the hearing impaired, investigators, or enable written archives. But annotating or searching through audio has become a very costly affair, requiring professional transcribers to spend hours listening to audio that might prove to be irrelevant. The time requirement is especially problematic in a forensics context as time can be of the essence.

In this master thesis we present a novel way to computationally find similar-sounding environmental sounds. We present a test data set that shows similarity between sounds, as well as *Sound2Vec*, a script to convert audio into short-form vectors that can quickly be compared against a database. Sound2Vec uses the image classifier ResNet and transfer learning to extract features.

We perform experiments on classification and similarity measuring and show a top-1 classification accuracy of up to 75%, a top-1 similarity accuracy of 22%, and a top-5 similarity accuracy of up to 55.5%. Each sound could be classified or compared in less than 200 ms.

Sammendrag

Hver dag tas det opp tusenvis av timer med lyd i form av lydopptak og video. Denne lyden må noen ganger kommenteres og transkriberes for å hjelpe hørselshemmede, etterforskere eller muliggjøre skriftlige arkiver. Men å kommentere eller søke gjennom lyd har blitt en svært kostbar affære, og krever at profesjonelle transkriberere bruker timer på å lytte til lyd som kan være irrelevant. Tidskravet er spesielt problematisk i en etterforskningsammenheng da tid kan være avgjørende.

I denne masteroppgaven presenterer vi en ny måte å automatisk finne lignende miljølyder. Vi presenterer et testdatasett som viser likhet mellom et sett med lyder, samt *Sound2Vec*, et program for å konvertere lyd til kortformede vektorer som raskt kan sammenlignes mot en database. Sound2Vec bruker bildeklassifisereren ResNet og transfer learning for å trekke ut features.

Vi utfører eksperimenter for å måle programmets klassifiseringsnøyaktighet og evne til å finne like lyder. Resultatene viser en topp-1-klassifiseringsnøyaktighet på opptil 75%, en topp-1 likhetsnøyaktighet på 22% og en topp-5 likhetsnøyaktighet på opptil 55.5%. Hver lyd kan klassifiseres eller sammenlignes på mindre enn 200 ms.

Acknowledgments

Completing this thesis would have been impossible without the guidance from my supervisor Lasse Øverlier and informal co-supervisor Kyle Porter. Thank you both for giving me feedback on my ideas and my writing!

I also want to thank Henrik Havnes for lending me his graphics card, without which the work would have been significantly hampered. Additionally, I thank both Henrik and Ole Martin Holm for being willing to proof-read my thesis and giving invaluable feedback.

Contents

Figures

Tables

Chapter 1

Introduction

There is very often a need to convert the contents of audio into written words. This is called *transcribing* audio and is either limited to speech or speech *and* sounds and musical descriptions. Transcribing has historically been to aid those who cannot easily perceive audio, such as those with hearing disabilities, as well as to archive the contents of audio. Though, because of the Watergate scandal in 1973, there was suddenly a need to transcribe, enhance, and process audio for the purpose of finding and presenting evidence. The field of audio forensics was largely established because of the investigation [1].

Audio can often be an important source of evidence in an investigation and performing audio forensics has therefore become more common. Audio forensics covers the acquisition, analysis, and evaluation of audio recordings for the purpose of potentially presenting them as evidence in court [2]. A part of this is to find sounds of interest and transcribe the audio, making sure to make reasonable conclusions about what each sound is.

One type of sound of special interest are environmental sounds. Environmental sounds are the sounds that describe events in the environment, such as barking or thunder. In audio forensics, these sounds can be a major source of evidence, as they can describe what happened without relying on someone speaking.

Though transcribing sounds can be an arduous process. The quality of audio can often be too low to clearly differentiate sounds and listening for specific sounds in noisy audio is tiring and time-consuming. It stands to benefit from automation.

1.1 Keywords

Audio similarity. Audio classification. Convolutional Neural Network. Environmental Sound Classification. Audio Forensics.

1.2 Problem Description

With large amounts of video and audio recorded every day, transcribing it all would be impossible. Transcription of speech can be largely automatic – just consider YouTube's automatic annotation or Google Translate speech-translation feature – but annotating environmental sounds, such as knocks, wind, and footsteps, can often be a manual and tedious process, depending on the level of accuracy required [2]. Annotation of speech and sounds are important for people with hearing disabilities, as well as for those who must record the contents of audio in an easily searchable text-format, such as archivists or the police. While the speech annotation can be semi-automated depending on the level of required quality, sounds still largely require manual listening.

The systems that do exist focus on class-level annotations [3][4]. That is, instead of being descriptive, they group sounds. For example, they say the sound bears resemblance to a gun, instead of a 9 mm caliber pistol. Certain classifiers try to be more descriptive [5], but they have to create data sets specific to the sound. Creating these data sets can be time consuming and labor intensive. These classifiers are therefore often limited to specific sounds, such as guns [5] and heart sounds [6].

An alternative to this approach would be to compare sounds to a database of reference sounds. Similar sounds are likely to be caused by the same event and can therefore be annotated using the same description. Humans can tell a lot about a sound based on similarity to sounds they have heard before. A similarity measuring system must therefore try to mimic how humans perceive audio as similar.

To this end, we present a system for comparing sound similarity to a database, as well as a testing data set based on human perception to test such a system.

1.3 Topics Covered

The focus of this thesis is on measuring similarity between environmental sounds. To this end, we will design a framework that extracts a set of features that can be compared to indicate similarity between two sounds. The purpose is to determine the event that caused the sound and/or its properties. In order to do this, we will look into the field of psychoacoustics [7] to determine how humans consider similarity, the fields of audio forensics and audio processing, as well as the field of environmental sound classification to determine how to quantitatively compare environmental sounds.

1.4 Research Questions

1. Can we develop fingerprints of sounds such that their similarity can be quantitatively compared? What would the matching speed and accuracy be?

2. Can image classifiers be used to extract similar features for similar sounds?
3. Can a measure of similarity be used to classify sounds, and/or to describe the sound?

1.5 Justification, Motivation, and Benefits

Annotation and finding sounds in audio are very relevant problems [4]. Most of the current sound recognition techniques work based on classes of sounds [3][8][4], limiting their applicability to high-level descriptions. We wish to annotate sounds based on their similarity to other sounds, using image similarity techniques. Some applications for such a system would be in video and audio editing to find similar sound effects; captioning; surveillance; health monitoring; bird recognition; and forensics.

In particular, audio forensics, a subset of digital forensics, is the context of this thesis. One of the important steps in the audio forensics process is to interpret and document audio-based evidence [2]. Investigators therefore have to dedicate hours combing through audio or hire experts to do it for them. While recordings used in courts often must be manually listened to, and audio annotation systems are generally combined with manual intervention to increase accuracy and speed, we hope an automatic audio similarity measuring system could alleviate some of this work by allowing investigators to either search after a specific sound, such as a gunshot from a specific gun, or annotate recordings automatically to highlight potential areas of interest.

1.6 Contributions

In this thesis we present a test data set that contains quantitative annotations that quantifies perceived similarity between a subset of sounds from the ESC-50 data set [9], as well as a feature extraction framework that allows for quantitatively comparing similarity. The framework is based on the ResNet-family of convolutional neural network, and is designed to output 128 features per sound, whose euclidean distance to similar sounds is short. Note that there are multiple types of similarity, and we focus on similarity related to detecting the event that caused the sound and its properties.

1.7 Outline

To answer the research questions, we start by discussing the relevant background and theory, such as literature regarding audio and auditory similarity, as well as related works from the fields of audio processing in chapter 2.

In chapter 3 we explain how this information was used to design a couple of studies to answer the research questions. Study 1, covered in chapter 4, explains how we created a data set that measures auditory similarity. Study 2, in chapter 5,

describe how we designed a fingerprinting framework called *Sound2Vec*, experiments for testing how discriminating the fingerprints were, and the results from these experiments.

We discuss the results in chapter 6. Potential future work and the thesis conclusion is presented in chapter 7.

Chapter 2

Background & Related Work

In this chapter we will introduce relevant theoretical background knowledge from the field of audio processing, and related works related to audio similarity.

2.1 Psychoacoustics and Audio Similarity

The definition of a sound depends on the field of study. In physics, it is a vibration propagating through a medium such as a gas. While in human physiology and psychology, sound is limited to human reception and perception of such waves. Thus, for a human, a sound is a wave passing through the air with frequencies between 20 Hz and 20 kHz. We use this definition in this thesis.

One of the fields concerned with these types of sounds is psychoacoustics. It is the field concerning human perception of sounds and audiology and is an interdisciplinary field. It draws from psychology, acoustics, physics, biology, physiology, electronic engineering, and computer science [7]. This is because these fields all touch upon human perception of sound at some point. Just processing digital audio in the brain requires a digital computer to output digital signals to a speaker, have the speaker convert the signals to waves in the air, received by a human ear, converted to electrical signals in the cochlea, and then transported to the brain and processed.

The field also concerns itself with human's auditory limitations [7]. Humans are fine-tuned to survive their environment, meaning we strike a balance between the benefits of hearing better and the energy costs of growing and maintaining complex hearing and processing mechanisms. Humans therefore have multiple perceptual limitations. Focusing on auditory perception, humans generally only perceive sounds frequencies between 20 Hz and 20 kHz, a range that degrades as we age. The body also has a non-linear relationship to most aspects of audio: perceived differences in loudness increase more slowly as the sound's intensity goes up, and changes in frequency are detected less as the frequency increases.

When it comes to audio, we often think of three different types: music, speech, and environmental sounds. Music is primarily acoustical and consumed for pleasure. Speech is meant to communicate complex thoughts and is limited to small

frequency ranges. Environmental sounds are processed to detect events in the environment, allowing humans to orient themselves.

In this work we focus on environmental sounds; commonly just "sounds". To differentiate our focus from the fields of speech and musical processing, we adopt the definition of environmental sounds as first presented by Vanderveer [10]. He defines environmental sounds as any audible acoustic event happening in a normal human environment, that:

1. has a real event as their source,
2. are more complex than laboratory sinusoids,
3. are meaningful, in that they specify the event that caused it,
4. are not intended for communication; they are considered in their literal interpretation.

The rest of this thesis will focus on these types of sounds and measuring similarity between them. The following subsection covers how humans perceive sounds and how they can effectively be compared. Following that, there is a short introduction to techniques most commonly used for environmental sound classification.

2.2 Auditory Similarity

The perception of similarity arises from what a listener focuses on. Experimentally, we know that subjects will focus on different things depending on what *mode* of listening they are doing. These modes are *musical listening* and *everyday listening* [11][12]. Musical listening has the listener focus on the qualities of the acoustic signal – the pleasantness, loudness, pitch, and so on; while a listener doing everyday listening tries to orient themselves in their everyday life, identifying the events causing sounds and their properties.

Though in reality, the separation of these modes are not clear. It has been shown that acoustical properties alone are rarely enough to determine the properties of the event. In situations where the audio is not clearly similar, listeners combine musical and everyday listening to group similar sounds [10].

When not doing grouping, and the listener is instead tasked with describing a sound, they will generally describe it using three attributes: the object making the noise; the action taken upon that object; and where the action took place [10]. For example, "A single wood plank dropped on concrete in a tunnel".

We can therefore observe three base strategies humans employ to measure similarity [13]. In some situations, sounds are grouped after (1) clear acoustical similarities or (2) a clear source, where little interpretation is needed. However, in some situations, it is necessary to identify precisely what caused the sound, who or what caused it, why it happened, and so forth. Being able to tell this information relies heavily on (3) a listener's knowledge of the sound and context [14][15][16][17].

This observation gives rise to three similarity types [13]:

1. **acoustical similarity**: similarity based on acoustical properties.
2. **causal similarity**: similarity based on the object/event and its properties. This is considered as analogous to classification.
3. **semantic similarity**: similarity based on some knowledge or meaning a listener attaches to the object/event causing the sound.

The focus of this thesis is on causal and semantic similarities. Any model capable of measuring semantic similarity should be capable of measuring causal similarity.

2.3 Methodologies to Measure Auditory Similarity

Even if sound similarity is well defined, it is still not easy to task someone with comparing sounds. While not an area of much study, how to measure similarity between sounds is very important to effectively and correctly measure human-perceived similarity. The most obvious way of doing it is to compare every sound against every other sound, but that is a very time-consuming and mentally taxing process. Another way is to extract a number of features and group the sounds based on those. However, this method has very little research [18], and requires selecting the correct features and method.

Here we will discuss the benefits and the drawbacks of these two techniques.

2.3.1 Pairwise Comparisons

Pairwise comparisons is a very intuitive way to measure similarity. Simply compare every sound with every other sound. Give high scores for similar pairs, low scores for dissimilar pairs.

But human perceptual, cognitive, and decision strategies are too limited to efficiently and accurately apply pairwise comparisons [19]. Listeners will commonly ignore the big picture, and instead compare based on the most prominent dimension, which can be acoustic, descriptive, or categorical [20]. Secondly, it can be expected that a listener will generate new criteria as they get to know the sound more [19]. The annotator should double-back and adjust their previous scores, which is taxing and can impact accuracy on future comparisons. Thirdly, it is difficult to keep the scale of similarity constant. A listener might consider the scale only in relation to recent sounds, making annotators fail to uniformly apply the scale [19]. The final problem with pairwise comparison lie in the rapidly growing number of comparisons needed for every new sound added to the dataset $(n * ((n-1)/2))$.

2.3.2 Grouping

An alternative is "grouping". In grouping, participants are presented with features that represent the sounds and told to group them based on similarity. For example,

presenting measures of pitch and tone allows grouping of urgency in nonvocal auditory warning signals [21]. The main benefit of grouping is that it allows sounds to be grouped without having participants perform multiple comparisons. This method is understandably faster and requires less from the participant.

However, grouping comes with several practical problems. Firstly, it relies heavily on using the "right" methodology, which differs between what sound is being compared [19]. Research also indicates that what features to present to an annotator differs based on their expertise [22]. What constitutes a similar sound also differs between groups of people, making it unlikely that a broad and general feature for similarity can be identified.

Further, and arguably more important, grouping is best for exactly that – grouping. It enables rapid classification of sounds based on feature similarities but does not necessarily measure similarity between sounds.

2.4 Audio Processing

There are many choices that must be considered when processing digital audio. How the audio is recorded, converted, processed, and how features are extracted can have a significant impact on a machine learning model's performance, or waste space and processing power by introducing redundant information.

To understand the choices made in this thesis, we introduce common options to consider in the process of converting raw digital audio into features, starting with sampling.

Sampling

Both humans and computers convert analogue audio into electrical signals. However, computers encode audio using an Analogue-to-Digital converter to transform it to 1s and 0s, while humans use the cochlea to convert it into neural action potentials [23]. The limitations of using bits instead of electrical signals means that the audio cannot have infinite precision and must be imperfectly recorded.

The analogue-to-digital conversion works by "sampling" the wave's amplitude at given time intervals [24]. How often sampling occurs is known as the *sampling rate*, or *sampling frequency* [25]. Higher sampling rates means the wave is represented more closely, but there is a point of diminishing returns.

The point of diminishing returns is commonly known as the *Nyquist Frequency*, described by the Nyquist-Shannon Theorem [26]. It states that if a discrete signal is sampled at least *twice the maximum frequency component*, the signal can be completely reconstructed. The maximum frequency component is the highest frequency in a signal, and the Nyquist Frequency is therefore the maximum frequency component. This means a wave must be sampled more often as the frequency goes up to maintain a similar resolution (see Figure 2.1).

Recording with a too low sampling rate can introduce *aliasing*, where the digital version of the signal becomes distorted. Any frequency above the Nyquist Fre-

quency will appear as lower frequencies, potentially introducing unpleasant noise when converted back to analogue. These frequencies are commonly filtered using filters such as low-pass or anti-aliasing filters, but the simplest and most effective solution is to increase the sample rate [27].

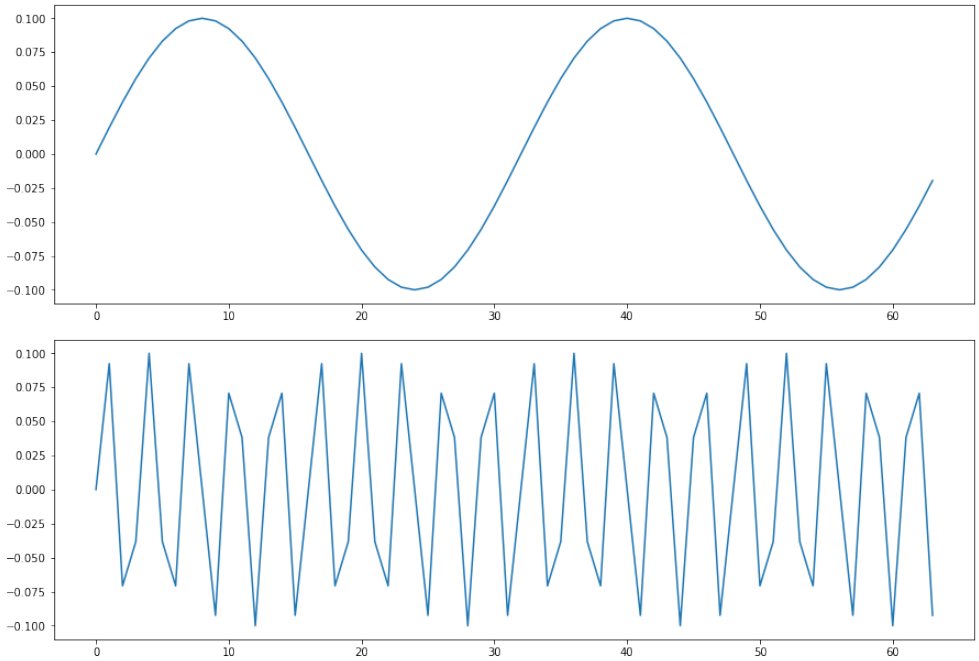


Figure 2.1: Sine wave at 500 and 5000 Hz with a sampling rate of 16000 Hz. Capturing the sound wave starts to break down at higher frequencies unless the sample rate is increased.

To prevent aliasing, it is common to sample audio signals at 44100 Hz. This number comes from a Nyquist Frequency of 22050 Hz, which adds some slack to a human's upper hearing limit of 20 kHz.

When the audio has been sampled and digitized, it might need further processing before being used in audio processing. Some processing techniques can work directly on the raw audio, but many others rely on decomposition. One common method to decompose the signal further is to extract its component frequencies by using the Fourier Transform.

Fourier Transform

The Fourier Transform (FT) exploits that all continuous and digital functions can be decomposed into a series of sine waves [28]. In other words, a complex waveform can be built up by, and converted back into sine waves. A sine wave is a pure tone and has a single frequency. Subsequently, we can determine frequencies that are present in a waveform by decomposing it into sine waves. These frequencies are called the component frequencies.

But the original Fourier Transform expects infinitely precise input, an impossibility on digital computers. As a solution, the whole-integer based Discrete Fourier Transform (DFT) was introduced.

However, using the definition of the DFT is computationally expensive, and most implementations use one of the Fast Fourier Transforms (FFT) instead. FFTs are significantly faster versions of the DFT that reduces the computational complexity from $O(N^2)$ to $O(Nlog_2N)$. Additionally, FFTs are more accurate than the DFT in the presence of round-off errors. One of the most common FFT algorithms is the Cooley-Tukey algorithm [29] and is one of the algorithms used by the python library `scipy`, on which libraries like `numpy`, `librosa`, and `pytorch` base their FFT implementation [30][31].

Short-Time Fourier Transform However, the Fourier Transform works on entire signals. This means it returns a list of all frequencies within that signal, without any information about when the frequencies occurred.

The Short-Time Fourier Transform (STFT) solves this issue by dividing the signal into equally sized chunks and calculating the transform over them. A window is a snippet of time – a continuous series of samples – that is extracted by zeroing the rest of the audio. This is generally done by multiplying the signal with a *window function*. There are multiple window functions, such as 'Hann', 'Hamming', and 'Blackman', but none are perfect. Due to how waves work, zeroing out sections introduces *spectral leakage*. Spectral leakage is the appearance of frequencies that do not exist within the signal. It cannot be eliminated and must therefore be controlled by selecting a fitting windowing function or increasing the window length.

Another downside of the STFT is that it does not provide exact frequencies. Instead, it gives multiple ranges of frequencies. This is called the spectrum/frequency resolution, and a range is often called a "bin". The resolution depends on the size of the window and the sample rate of the signal: $resolution = \frac{sample\_rate}{window\_size}$. Increasing the window size increases resolution, but also increases the time between when changes can be detected. For example, at a sample rate of 44.1 kHz and a window size of 4096 samples, the STFT has a resolution of 10.77 Hz every 0.1 second. Meaning, we cannot tell the difference between a frequency at 17.15Hz and 24.03Hz because they are within the same bin, nor is it possible to detect changes in the frequencies faster than every 0.1 second. Changing the window to 16384 samples gives a resolution of 2.7 Hz every 0.37 seconds. An imperfect method to allow larger windows is to overlap the windows, allowing changes to be identified more often, but with less certainty of where frequencies occur and no increase in resolution.

Spectrograms

The STFT returns a spectrum of the signal: A 2-dimensional matrix, where frequency resolution and time are the axes, and the values are amplitude. We are

interested in the energy or power of the signal, so we take the absolute of the STFT and square it (see Equation 2.1).

$$power\_spectrum = |STFT|^2 \tag{2.1}$$

But the STFT has a small number of large values, as well as a significant number of small values. This means we throw away a lot of information. We therefore convert the power spectrum into the decibel scale (see Equation 2.2).

$$spectrogram = log_{10}(power\_spectrum) \tag{2.2}$$

This provides a spectrogram.

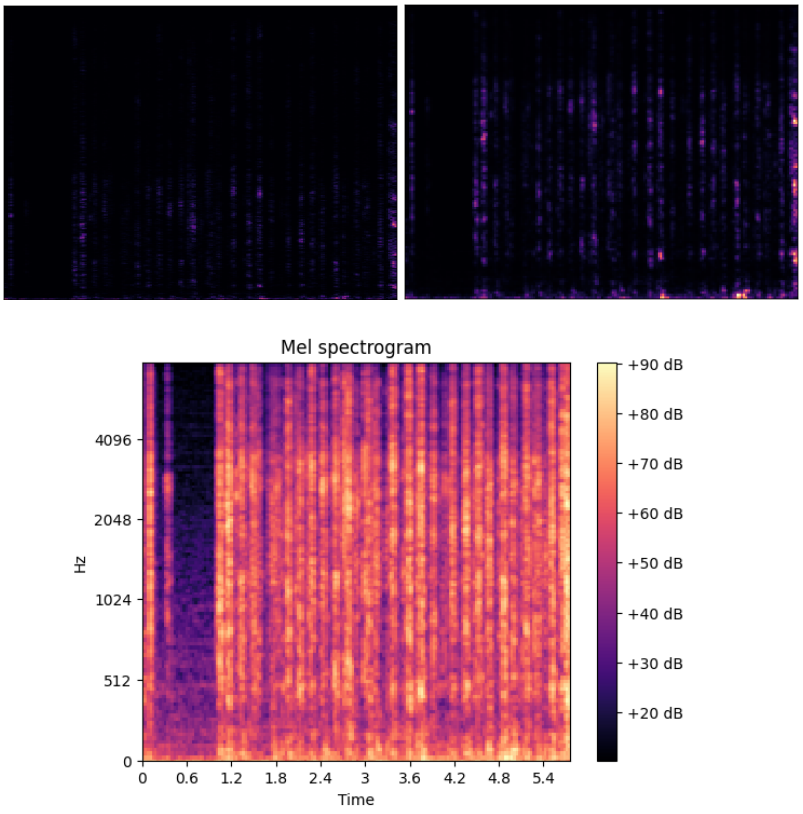


Figure 2.2: (left to right, top to bottom) An STFT spectrum, power spectrum, and a mel spectrogram of the same recording. The difference between a spectrogram and a mel spectrogram is just the scale.

Mel Scale

While a spectrogram is an accurate representation of an audio signal, it is a poor representation of pitch. Pitch is a musical term describing the human perception

of frequency. It's been experimentally verified that humans find higher frequencies more difficult to differentiate than lower ones. And a spectrogram weighs all frequency equally.

To address this issue, a group of researchers at Harvard and Swarthmore introduced the Mel-scale in 1937 [32]. The Mel scale, named after the word "melody", is a subjective scale measuring perceived similarity of pitches. It shows that human listeners need increasingly large increments in frequency to notice changes as the frequencies go up (see Figure 2.3). Specifically, above 500 Hz, increasingly large intervals are described by listeners to give equal pitch increases. A "mel" is a mapping between a frequency and *perceived* frequency. The scale is designed such that 1000 mels are the same as 1000 Hz.

Note that this scale is considered flawed, being most likely significantly biased. One of the creators' students publicly criticized the methodology, citing the few participants (five) and lack of bias control [33]. However, it is still commonly used and is probably a decent approximation of human perception.

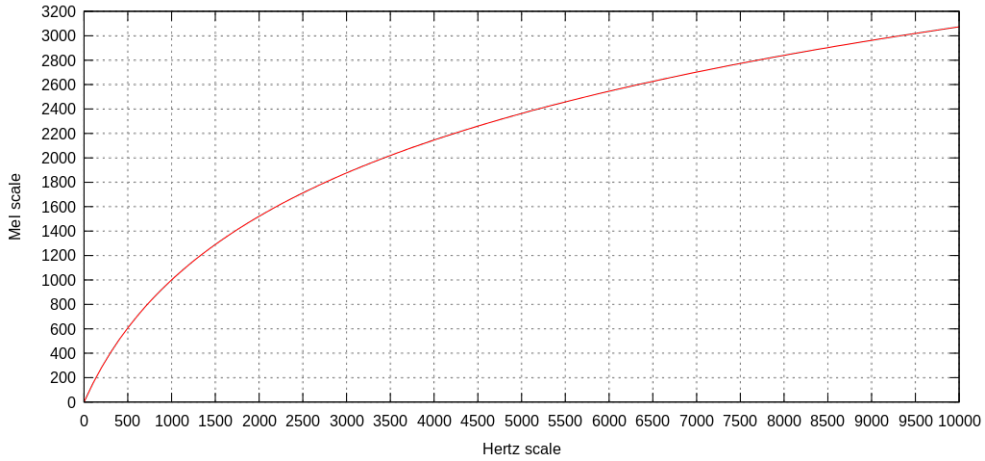


Figure 2.3: Pitch on Mel scale versus Hertz scale [34]

Spectrograms can be converted to Mel-spectrograms, where frequencies are spaced according to the Mel scale. This is done by mapping a spectrogram's frequency bins into mel bins. Mel-bins being several frequency bins with increasing size as the frequency goes up according to the mel scale. Notice how the mel-spectrogram's frequency scale in Figure 2.2 are non-linearly spaced. The user specifies the number of bins, meaning the size of a mel-spectrogram in the y-dimension is independent of its window size or sample rate.

2.5 Related Work

There are many different fields in audio processing, and it is generally recognized that most algorithms from these fields will at some point measure some kind of

similarity [13]. Some of these fields are music information retrieval, speech recognition, instrument recognition, and environmental sound classification. Of course, these fields consider different types of similarity. Music-oriented fields consider acoustical similarity, and the others often consider some form of causal and semantic similarity.

As far as the authors of this thesis know, there are no research papers where similarity between two environmental sounds are computationally measured – at least not for semantic purposes. Music Information Retrieval almost exclusively works on comparing similarities between audio, but music is too ordered to allow the transfer of techniques from that domain. Instead, this thesis relies upon the closest field that considers some similarity between environmental sounds: environmental sound classifications (ESC).

Up until recently, it has been very common to process environmental sounds by special purpose algorithms, using specially crafted features [3]. This led to highly domain-specific algorithms and made it difficult to benefit from advances in other fields, such as computer vision.

However, at some point, vision models seemingly became the de facto standard in sound processing [8]. More specifically, convolutional neural networks (CNNs) became very common – both networks purpose-built for audio or simply transferred from vision tasks. [1]

In this section we will present some of the related work in the field of environmental sound classification (ESC). First, we present the state of data sets and common features, then we briefly present early methods of ESC, before moving on to current deep learning methods. The remaining part of the section covers the details of the ResNet convolutional network and how it relates to audio classification.

2.5.1 Data sets

Historically, there has been a lack of universal data sets for environmental sounds. Some domain-specific sets did exist, but the sets were too limited in scope [35]. This often led to papers using their own data sets, with an arbitrary number of samples, of varying quality, that were not easily available.

However, in recent years three universal data sets have become the de facto standard: UrbanSound8K, Google's AudioSet, and ESC-50/-10. They are universal in the sense that they cover a broad range of sounds and are sizeable. These data sets are used for classification purposes and therefore represent causal similarity. As far as the authors know, a data set that measures semantic similarity does not exist.

UrbanSound8K was released in 2014 and has 8732 labeled sound excerpts at 4 seconds or less [36]. It has ten classes, all representing common sounds present in an urban environment. It became popular due to being pre-sorted into 10 folds,

[1] Models evaluated on ESC-50. 28 out of 33 papers with scores over 70% accuracy use CNNs: https://www.github.com/karolpiczak/ESC-50

allowing cross-validation to be effectively compared between research articles, and because it was a large data set that was hand-labeled. Its major drawback is how noisy the samples are.

Google's AudioSet is an ever-expanding ontology of sounds drawn from YouTube videos [37]. The data set is huge, numbering some 2.1 million samples, but the quality of the samples varies wildly. Its claim to fame is the sheer number of classes and samples it has, allowing any researcher to create data sets specific to their needs. But due to not being pre-defined, it is difficult to compare results between research papers. Another problem is that the raw waveforms are not released and must be extracted by the researchers.

ESC-50/-10 was released in 2015 and is very commonly used [38]. It is a collection of 2000 hand-labeled samples, spread over 50 classes with 40 samples each. Every sample is 5 seconds long, meaning little to no pre-processing is necessary to standardize the samples. Compared to UrbanSound8K, it has more classes and a wider range of sounds, but only five pre-defined folds. Additionally, a list of results on models evaluated on ESC-50 is available online.

ESC-10 is included as a subset of ESC-50. It includes 10, easily separable classes, so the classification accuracy is generally higher. It is only expected to be used as a proof-of-concept data set but is commonly included in the research results.

Features

Features used in deep learning ESC are commonly extracted in a way that considers time. More often than not, the features are STFT spectrums or derivatives of them, such as spectrograms, mel-spectrograms, and cepstrum coefficients.

Since CNNs are like vision models, the features presented to them have commonly been 2-D matrices, mimicking an image. There is much research on what these features should be, but there are mainly three common ones: spectrograms, cepstrums, and Cross Recurrence Plot (CRP).

Spectrograms are by far the most commonly extracted features. In order of popularity, there is the mel-spectrogram, normal spectrograms, and the Gammatone-like spectrogram. The mel-spectrogram can be seen in a multitude of research, often achieving high classification accuracy on ESC-50 [39][40]. However, this is also true for both normal and GammaTone-like spectrograms [35][41]. They often get the same or similar scores depending on the algorithm. Lacking a comprehensive review of the features, it is difficult to say what spectrogram is the optimal feature for the final accuracy.

Mel-Frequency Cepstral Coefficients (MFCC) are also common. Cepstrums are designed to mimic the human auditory system and have been successfully applied in speech and music applications. MFCCs are a decomposition of spectrograms and can be presented in a equivalent way to spectrograms. They often have high classification accuracy in the literature [39][42]. However, MFCCs are sensitive to noise, and might not apply well outside of limited research datasets

[43].

Cross Recurrence Plot (CRP) is a less common feature. A CRP is a matrix visualization of the distance between phase trajectories of a time series, such as an audio sample. However, this one is by far the least used feature, and indications point to it not performing as well [42].

2.5.2 Early Methods for Audio Classification

Early methods for environmental sound classification rely heavily on aspects imported from speech and music recognition [3]. Common features have therefore been psychoacoustic properties, such as loudness, pitch, and timbre, which were grouped based on machine learning models such as K-nearest neighbors, Gaussian Mixture Models (GMM), and support vector machines [44][45].

But these features generally rely upon the audio being stationary, with the same acoustical properties throughout. Audio, and especially the more chaotic environmental sounds, often violate this principle.

To address this, a number of research papers considered using non-stationary techniques, relying on time-sensitive features such as spectrograms, Mel-Frequency Cepstral Coefficients (MFCC), Wavelets, and Matching Pursuit [3][46]. But these techniques were still more-or-less imported from music and speech processing, often relied on simple artificial neural networks and GMMs, and performed poorly.

2.5.3 Deep Learning

Following the advent of larger, standardized data sets in 2015, papers using deep learning techniques quickly became more common. Deep learning is a class of machine learning algorithm, where multiple neural network layers are used to extract increasingly complex information from input data [47]. The layers allow the algorithm to simulate a brain and learn from data. All new environmental sound classification techniques have been using some form of deep learning, of which Convolutional Neural Network (CNN) are the most popular [8].

CNN are a type of deep neural network, where so-called convolutional layers are included. The idea is that these layers allow the network to recognize more complex features the further down the stack it goes. CNNs have primarily been used as image recognition tools. The first layer identifies edges, the second identifies shapes, the third identifies objects, and so on.

One of the earliest papers on using two dimensional CNN for the task of environmental sound classification was introduced in 2015 [48]. The model used its own network architecture and Mel Spectrograms as features. Importantly, it set the first baseline for on the ESC-50 data set at 65.5% accuracy, beating previous techniques using SVM and k-NNs by a significant margin [9]. Later studies combined 1-D convolutional layers with fully connected layers to extract features from the raw waveforms, achieving 71.0% accuracy on ESC-50 [49].

The paper spurred much research on using convolutional neural networks for the purpose of environmental sound classification [35][42][50]. Many different

CNN architectures were created in the beginning, sometimes pre-trained on AudioSet [51] or a general audio data set [52].

ESC-50's GitHub page lists many recognized models, their accuracy when evaluated on ESC-50, and a link to the relevant paper.[2] In Table 2.1 we present a select few papers, including a few results from other sources. This includes the state-of-the-art, the ESC-50 baselines, and a number of entries using CNNs. Refer to Table 2.1.

Table 2.1: Select classification results from the literature.

Paper Title	**Description**	**Accuracy**
AST: Audio Spectrogram Trasnformer [38]	Pure Attention Model Pre-trained on AudioSet	95.70%
A Sequential Self Teaching Approach for Improving Generalization in Sound Event Recognition	Multi-stage sequential learning with knowledge transfer from Audioset	94.10%
Efficient End-to-End Audio Embeddings Generation for Audio Classification on Target Applications	CNN model pretrained on AudioSet	92.32%
Fine-Tuning ResNet-18 for Audio Classification [53]	Transfer Learning of FastAI's ResNet18	89.54%
Baseline – Human Accuracy [9]	Crowdsourcing experiment in classifying ESC-50 by human listeners	81.30%
How to normalize spectrograms [54]	FastAI's ResNet18 trained from scratch	73.15%
Environmental Sound Classification with Convolutional Neural Networks - CNN baseline [48]	CNN with 2 convolutional and 2 fully connected layers, mel-spectrograms as input, vertical filters in the first layer	64.50%
Baseline - k-NN [9]	Baseline ML approach (MFCC & ZCR + k-NN)	32.20%
Baseline - SVM [9]	Baseline ML approach (MFCC & ZCR + k-NN)	39.60%

However, the limited size of the data sets was limiting the depth the models could be. At some point, researchers realized that performing transfer learning on CNNs pre-trained on image data sets such as ImageNet [55] performed well for environmental sound classification [35][53]. Transfer learning is a method in machine learning where a model trained for a task is reused as the starting point for a second task. Commonly, in environmental sound classification, models have

[2]https://github.com/karolpiczak/ESC-50

been created for image classification.

Image classification models such as GoogLeNet [42], AlexNet [42], ImageNet[3] [50], ResNet [53], and DenseNet [56] have all been used to classify sounds, achieving, at some point, state of the art performance. ES-ResNet-Attention [35] extends a ResNet50 model pre-trained on ImageNet and achieves 91.5% accuracy on ESC-50. Table 2.2 summarizes these models, as well as their highest accuracy.

Table 2.2: Highest accuracy for image recognition networks evaluated on ImageNet and ESC-50.

	AlexNet	**GoogLeNet**	**MobileNet**	**ES-ResNet**	**DenseNet201**
ImageNet Top-1 Accuracy	56.55	69.78	71.88	69.76	77.65
ESC-50 Top-1 Accuracy	68.70	73.20	90	91.5	92.89

Conversely, the state-of-the-art in environmental sound classification is the convolutional-free "AST: Audio Spectrogram Transformer" [57][58]. It claims to be the "[...] first convolution-free, purely attention-based model for audio classification with support for variable length input [...]" [57]. According to their paper, they achieve 95%+ accuracy on different data sets covering environmental sounds and speech. There are papers that claim higher performance [59], but this paper achieves the highest accuracy without tweaking the model specifically for the data set [8].

However, this paper is, as they state, the only one of its kind. While transformer-based models are becoming more popular, most other recent papers on environmental sound classification are using CNNs. We choose to use Convolutional Neural Networks due to their prevalence in the literature.

2.5.4 ResNet

More specifically, we focus on the ResNet architecture. ResNet was chosen over DenseNet [56] due to being significantly smaller and easier to train, while achieving similar environmental sound classification scores. Additionally, ResNet for audio has become the de-facto standard outside of research and is known to perform well [53][54][60].

Microsoft proposed the ResNet family of CNNs in 2015 to solve the problem of training deeper models [61]. Before this, most networks performed worse with more layers, generally only performing best at 16 to 30 layers [61].

The solution was to introduce the "Residual Block", which consists of two or more convolutional layers. The residual block introduces a "shortcut", where output from the layer before the block is injected into the output of the block, enabling training of deeper networks. It also reduces the number of network parameters significantly, speeding up training. Figure 2.4 shows the general architecture of a block against a "normal" two-layer block.

[3]Not published as an article. Results are from `https://www.tensorflow.org/tutorials/audio/transfer_learning_audio#split_the_data`

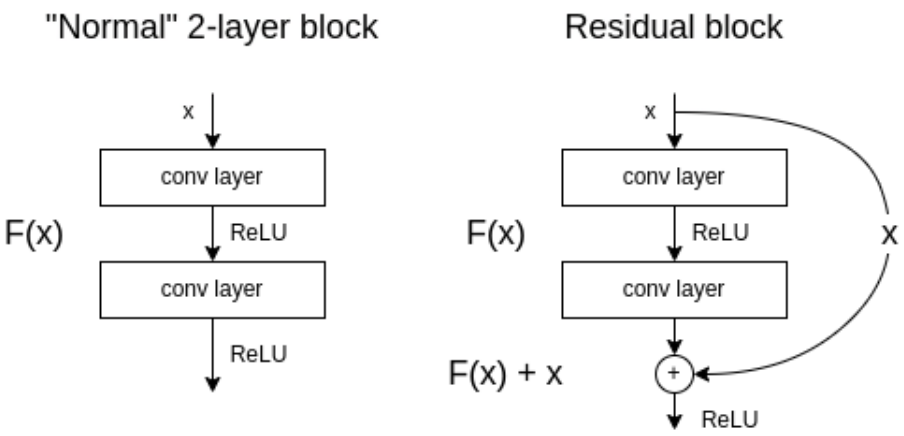


Figure 2.4: A two-layer "normal" conv block vs a residual block. The "X" is multiplied with a linear projection to ensure it is the same size as the output of the block. Figure is based on figure 2 in [61].

In the original paper, five network architectures are proposed with increasing number of layers [61]: 18, 34, 50, 101, and 152. The networks are defined by repeating the Residual Blocks or adding more convolutional layers into each block.

However, for the audio classification task, the smaller network architectures perform better. We therefore focus on ResNet18 and ResNet34, which have 18 and 34 layers each. See Figure 2.5 for the architecture of these networks. These two networks have a different residual block from the deeper networks but is otherwise the same. The networks output probabilities for 1000 classes by default, as the network is designed for the 1000-class ImageNet data set.

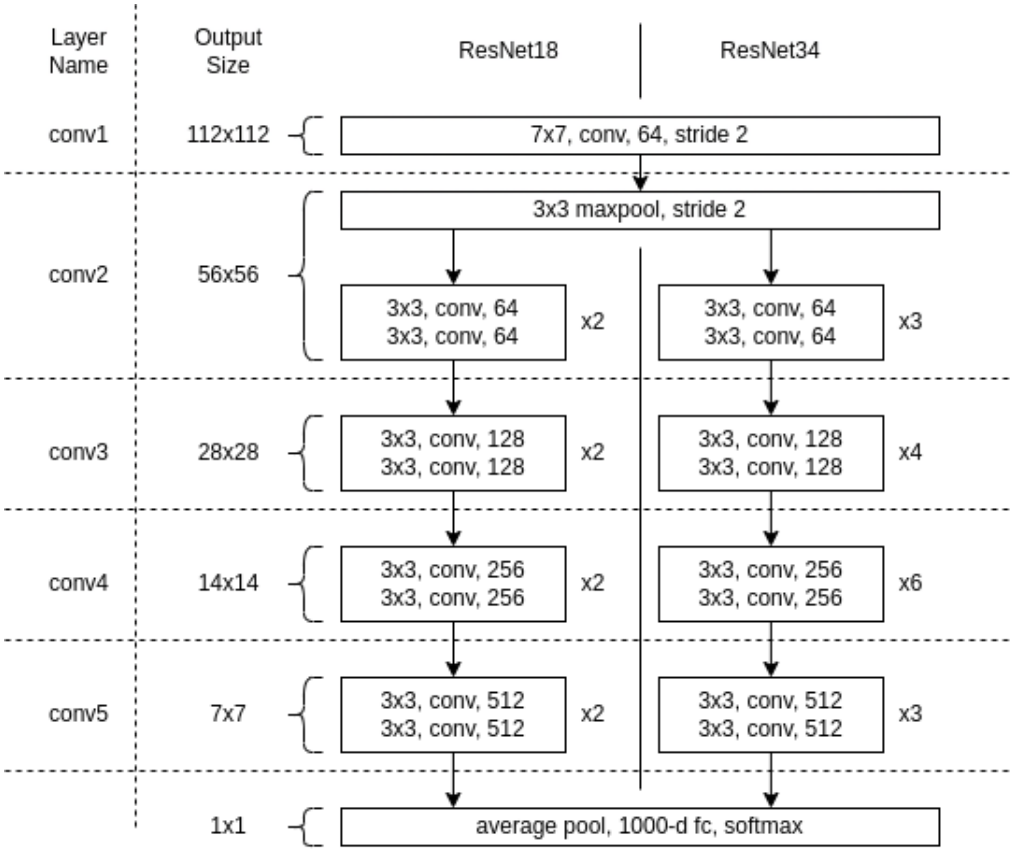


Figure 2.5: The architecture of the ResNet18 and 34 architectures. "xN" means the block is repeated N times. Output is 1000 values scaled to sum up to 1 (highest value indicates class). 1000-d fc is "1000 dimensional, fully connected. Softmax scales the sum of the values to between 0 and 1.

Chapter 3

Methodology

This thesis tries to answer if we can develop fingerprints of sounds such that their similarity can be quantitatively compared. The background shows that there are multiple types of similarity: acoustical, causal, and semantic. The focus of this thesis is on causal and semantic similarity due to them describing the event causing the sound. It should be noted that a model capable of semantic similarity should also be able to manage causal similarity.

This thesis proposes a feature extraction model and test data set to answer the research questions. The model extracts several features from a sound file, while the data set is used to evaluate the model's performance at measuring semantic similarity. Its causal similarity is measured using the ESC-50 classification data set.

Two studies were designed to create the data set and feature extraction model. This chapter presents an overview of the methodology, while specific details regarding studies 1 and 2 can be found in chapter 4 and chapter 5, respectively. The code and .csv files for the data set is available on GitHub.[1]

3.1 Study 1: Test Data Set

A review of the literature shows that data sets that measure semantic similarity do not exist. A new data set is therefore required to answer the research questions.

We chose to create a test data set based on ESC-50 in this thesis. Gathering and structuring a data set from scratch is a huge undertaking, with multiple considerations and difficulties. To simplify the work in this thesis, we chose to annotate a subset of the ESC-50 data set [38]. This data set is considered to have high-quality recordings, with accurate labeling. Additionally, it has small enough classes that can be annotated quickly; it has a broad range of different sounds; the audio is clean and free of noise; and, because of its common use in research, allows us to compare our results. The data set is limited to a test set, because annotating a larger data set would take too much time.

[1] https://github.com/Legwarmer2584/Measuring-Auditory-Similarity

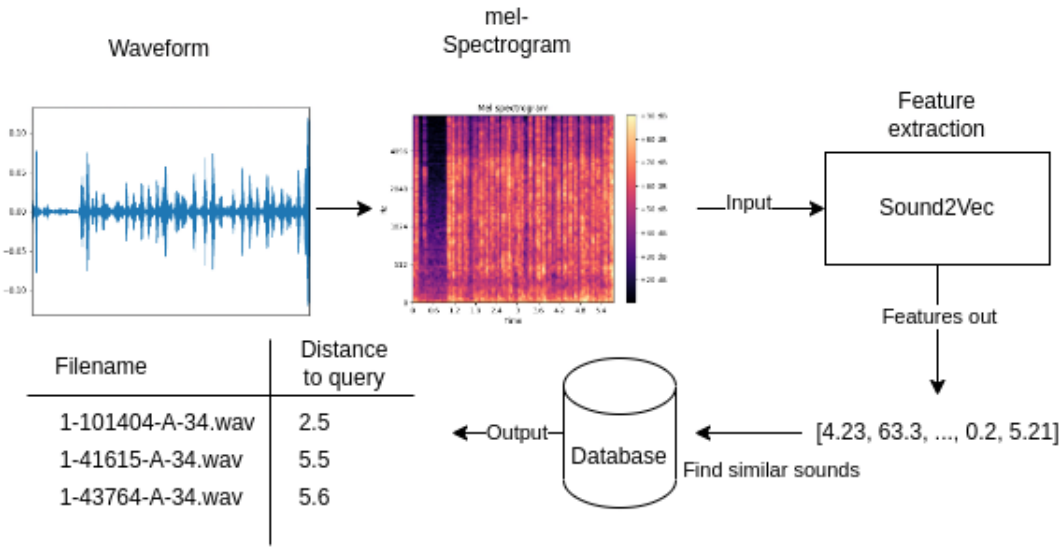


Figure 3.1: Description of the model.

Our annotated portion of the data set consists of five different sound classes. The sounds were selected to ensure that the listeners were likely to know the source and properties of the sound, as well as how a listener would primarily separate the sounds.

We chose to employ the pairwise comparison methodology, as described in subsection 2.3.1. This is appropriate, because the thesis focuses on having a deeper understanding of sounds than grouping can provide. Also, the lack of research on grouping makes it difficult to design the data set around it.

3.2 Study 2: Sound2Vec

The purpose of Sound2Vec is to take a spectrogram and output a vector, such that two vectors from similar sounds have a short distance between them. The short distance should then enable the recovery of similar sounds in a database, following the process shown in Figure 3.1. The distance should be a representation of how similar the two sounds are.

To achieve this, we convert a ResNet image classification model to do feature extraction on audio. The details of which are presented in section 5.1 and section 5.3. We use the two shallowest ResNet models, ResNet18 and ResNet34, and perform transfer learning on them. ResNet18 networks pretrained on ImageNet are proven to perform well on sound classification tasks after performing transfer learning [53]. Out of all the ResNet models, ResNet18 is often the most accurate at classifying ESC-50 [42][53]. We include ResNet34 to test if a more complex model can detect more subtle information useful to identifying similar samples.

We have chosen to use the Mel Spectrogram in this thesis. It is known to per-

form decently well at the audio classification task and is therefore a low-risk feature compared to MFCC, where robustness against noise is still a problem.

We train the model on the normal ESC-50 dataset. This is because our dataset is not sizable enough for the model to train on. Instead, we make the model output similar features for samples of the same class, providing the model an "understanding" of the sounds. Our hypothesis is that the understanding can be used to differentiate between sounds of the same class.

3.3 Performance Measures

Both ESC-50 and our data set are balanced with 40 sounds per class. For ESC-50, the cost of misclassification is the same for every class, and accuracy is therefore an appropriate metric when doing classification. The same goes for our data set when doing classification.

However, when finding the most similar sound, our data set can have many equally similar sounds. Meaning, multiple answers can be correct. Though, we still find accuracy appropriate as finding the most similar sample(s) is very similar to a classification task.

We employ the two accuracy-metrics "top-1" and "top-5" accuracy. "Top" here being a list of the best matches sorted after distance from the query. More precisely:

- **Top-1 accuracy**: The closest match is the right match.
- **Top-5 accuracy**: The right match is among the top five closest matches.

Chapter 4

Study 1: Dataset

There are no data sets that measures similarity. We also realize that creating a data set large enough to train on is unfeasible in the limited timeframe of the thesis. We therefore create a limited data set that can be used to test our theory.

The dataset is based on ESC-50 [9]. It has 50 classes with exactly 40 samples per class and comes with a pre-defined cross-validation split. The split has five folds, and we used the first fold as our validation set. We ran the experiments with the other folds after finding a network design and set of hyperparameters that achieved high accuracy on the validation set.

But ESC-50 is too large to be annotated in a reasonable timeframe. We reduce it by selecting five classes and annotate similarity between the samples in the same class. Below we cover why certain classes were chosen, and afterwards we cover our measure of similarity and how similarity was determined.

4.1 Sound Classes

Having multiple sound-classes to choose from, we selected sounds that are similar in diverse ways. We chose sounds where similarity depended on rhythmic development, frequency, and timbre. Another aspect was how much information the sound contained, and how well the annotators know the sound. Below, we describe the five chosen sounds and explain why they were chosen.

4.1.1 Soda Can Opening

A soda can opening is a complex sound that can originate from a wide range of possible can types. It can be indistinguishable from opening a generic aluminum can, except for a fizzing sound highlighting the soda contents. The most recognizable sound is from a 'wide-mouth' can using the 'Stay-Tab' opening mechanism, characterized by the two-step pierce-push sound (second on the third row in Figure 4.1). But 'Pull-Tabs', 'Topless', 'can-piercer' and 'Push-Tabs' mechanisms also exist. These all make their own distinct sound, and some are present in the data set.

Another factor impacting the sound is the person opening the can. Humans can drag out a sound, stopping to listen to the fizzing sound as the lever pierces the tab; they might add sounds in between by toying with the lever; and so on. The point is that the sound is unpredictable.

This of course does not take into consideration the shape of the can, which can give a completely different timbre and reverb.

Figure 4.1: Different soda can tops [62]

Perceived similarity therefore relies on not only on strong striking sounds; but also soft, drawn-out sounds; timbre; and reverb. Many different properties that will be difficult for a model to highlight.

4.1.2 Dogs

The dataset labels this sound as "dogs" but is better described as dogs barking. Meaning not other kinds of vocalization, such as howling, whining, and growling.

As opposed to those sounds, acoustic signals in barking can be very obvious. This makes sense, as dogs bark to tell us humans about their internal state [63]. Different frequencies, tones, and rhythms provide information exclusively for humans, with no proof of inter-species use. Subsequently there are strong patterns that can be recognized by a listener. However, there is a huge variability in size, anatomy, and species. Differentiating the meaning from similar-sized dogs on frequencies and tones alone can be difficult without knowing the species of dog.

But for most listeners, frequency identifies the size of the dog and barking rhythm is very noticeable. Similar sounds therefore have the same rhythm and similar frequency.

Finding similar sounds can therefore be considered "easy". Striking sounds are normally obvious peaks in a spectrogram, and a computer-vision model should be able to identify the patterns and frequencies than more complex auditory attributes. Especially for a model originally trained to identify objects.

Additionally, because barking follows patterns, there are multiple similar sounds even in the small dataset, increasing the chance that a good match is found.

4.1.3 Thunderstorm

The dataset uses a loose definition of a 'thunderstorm'. More accurately it is just 'Thunder'. The thunder sound is either a low rumbling or a strong strike of differing frequencies. Common backdrops are rain, crickets, silence, or rumbling from lingering thunder.

It is difficult for non-experts to extract information from the clips. Distant thunder reduces to indistinguishable rumbling. Close thunders are too intense to

differentiate on anything but frequency. Similarity boils down to the differences in frequency and backdrop.

Middle-distance thunders, however, are differentiable on patterns.

This class is included because of just that; There is a significant range on why two recordings are considered similar.

4.1.4 Church Bells

The church bells in the data set are from Christian churches, recorded from the outside. Conspicuously, a number of recordings sound like grandfather clocks, but we keep them in to keep the balance of the data set and comparability to other results.

A church bell is an instrument. They are tuned to play specific tones and the sound is musical in nature. Complex melodies can be played with one or more bells in a process called "change ringing". A well-known example of musical use is the bells of Great St. Mary's in Cambridge - an example the data set has multiple samples of.

In non-musical uses, most bellringers or automatic systems follow a rubric or a set interval. The page to play from, and the interval to keep depends on the occasion.

This class was chosen because it skirts the line between music and environmental sounds.

4.1.5 Pouring Water

The sound of pouring water comes from pouring water out of a container and into another container or onto a flat surface. Most people know the sound well, and it has significant harmonic content. A listener can therefore often provide much information about the water type and objects involved. High pitch is tied to colder water; lows with warm water; a tinny sound indicates cold water poured into a metallic container; and so on.

This sound stands out as having almost no striking sounds. Instead, it is smooth and noisy; covering a broad range of the frequency spectrum.

This class therefore highlights the model's ability to discriminate frequencies, ignore noise, and identify structures in the sound.

4.2 Similarity Scale

We chose to indicate auditory similarity on a scale from 1 to 5. A shorter range was used to give a wider margin of error and minimize the need to re-compare sounds. A higher score indicates more similar, with five meaning almost identical and one meaning completely different.

The application of the scale differs between classes. This is because of the lack of consistency over time.

4.3 Experimental Procedure

Two experiments were carried out. The first is the creation of the data set itself, and the second creates a baseline by randomly sampling similar samples.

The dataset was annotated by a single listener. This was done to reduce the time needed to create the dataset.

The audio was labeled using a scale from 1 to 5. A low score indicates dissimilarity, while high scores indicate high perceived similarity. This scale was chosen to reduce dependence on the listener's expertise, as the scale allows annotation when the listener does not know enough to label the sound (by using acoustical similarity).

The listener was presented with the audio in a pairwise manner. Pairwise comparison is the most intuitive method available. Simply compare every sound with every other sound. We gave high scores for similar pairs, and low scores for dissimilar pairs. This method was chosen because it is simple to implement and reduces dependence on listener expertise.

Since pairwise comparisons require $n*(n-1)$ comparisons in the worst case, we took a few steps to reduce the workload. First, we only compared sounds in the same class. This reduces n from 200 total to 40 per class. This was accepted because sounds of the same class were assumed to be the most similar. Secondly, to reduce comparisons further, it was decided to not compare test-samples from the same prearranged fold, reducing n to 32. These are instead annotated with '0'. Additionally, every sound only needs to be compared once, reducing n further to $n/2$. The final number of comparisons per class was therefore $(40-8)*(40/2) = 32*20 = 640$.

The completed data set is structured as a matrix and stored in the `csv` format. The file names are along the axes and the similarity score is in the intersection of the rows and columns. See Figure 4.2.

4.3.1 Hardware and Software

The audio was presented as mono, with a 16-bit resolution and a sampling rate of 44.1 kHz using the following hardware and software:

Purpose	**Hardware/Software**
Digital-to-analog converter & Audio Power Amplifier	Builtin DAC & AMP in ASUS ROG Strix B450-F Gaming
Headset	Philips Fidelio X3 headset (2020)
Operating System	Fedora 35
Audio Playback Software	Gnome Video 3.38.2
Audio pipeline	PipeWire 0.3.50

```
filename,1-101404-A-34.wav,1-41615-A-34.wav,1-43764-A-34.wav,1-58846-A-34.wav
1-101404-A-34.wav,0,0,0,0,0,0,0,0,2,3,2,2,1,3,3,2,2,2,1,2,2,1,3,4,3,3,2,3,2,2
1-41615-A-34.wav,0,0,0,0,0,0,0,0,3,2,2,1,1,2,3,2,2,2,1,1,2,1,2,4,3,4,3,3,2,2,
1-43764-A-34.wav,0,0,0,0,0,0,0,0,1,2,3,2,1,2,1,1,3,3,3,1,1,1,1,1,2,2,3,2,2,1,
1-58846-A-34.wav,0,0,0,0,0,0,0,0,2,2,2,1,1,2,3,3,2,2,1,2,3,2,3,3,4,3,2,3,3,2,
1-60676-A-34.wav,0,0,0,0,0,0,0,0,2,1,1,2,1,2,3,2,2,2,1,2,2,1,2,2,4,3,2,3,3,2,
1-68670-A-34.wav,0,0,0,0,0,0,0,0,2,1,1,1,1,2,2,2,2,2,1,2,2,1,2,2,4,3,2,2,3,2,
1-68734-A-34.wav,0,0,0,0,0,0,0,0,2,1,1,1,1,2,3,2,2,2,1,2,2,1,2,2,4,3,2,2,3,2,
1-69165-A-34.wav,0,0,0,0,0,0,0,0,2,3,3,1,1,2,2,1,2,2,3,1,1,1,2,4,3,2,2,2,2,1,
2-130245-A-34.wav,2,3,1,2,2,2,2,2,0,0,0,0,0,0,0,3,3,1,2,2,1,2,3,3,2,2,2,1,1
2-144031-A-34.wav,3,2,2,2,1,1,1,3,0,0,0,0,0,0,0,1,1,2,1,3,2,1,4,2,1,1,1,2,1
2-81112-A-34.wav,2,2,3,2,1,1,1,3,0,0,0,0,0,0,0,2,1,2,1,2,1,1,3,2,1,3,3,1,1,
2-81190-A-34.wav,2,1,2,1,2,1,1,1,0,0,0,0,0,0,0,3,2,1,1,2,3,2,3,1,1,1,1,1,1,
2-83667-A-34.wav,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,1,1,1,2,2,3,1,1,1,1,2,1,1,1,
2-83688-A-34.wav,3,2,2,2,2,2,2,2,0,0,0,0,0,0,0,2,1,3,1,3,2,1,3,1,1,3,2,1,1,
2-85471-A-34.wav,3,3,1,3,3,2,3,2,0,0,0,0,0,0,0,2,2,1,2,1,2,2,3,2,2,4,2,2,1,
2-87282-A-34.wav,2,2,1,3,2,2,2,1,0,0,0,0,0,0,0,1,2,1,1,1,3,1,3,2,2,3,1,1,1,
3-147342-A-34.wav,2,2,3,2,2,2,2,2,3,1,2,3,1,2,2,1,0,0,0,0,0,0,0,3,2,3,2,1,1
3-147343-A-34.wav,2,2,3,2,2,2,2,2,3,1,1,2,1,1,2,2,0,0,0,0,0,0,0,3,2,3,2,1,1
3-148932-A-34.wav,1,1,3,1,1,1,1,3,1,2,2,1,1,3,1,1,0,0,0,0,0,0,0,1,2,3,1,1,1
```

Figure 4.2: Example of the layout of the dataset csv file. Here from the "can opening" class. The top row shows the first four sounds; it extends far to the right.

4.4 Data Set Statistics

Overall, every sound had at least one sound that was more similar than a score of 1. Most of the sounds' most-similar counterparts had a score of either 3 or 4, with very few only having 2 or 5. Statistics over the distribution of scores within each class is presented in Table 4.1.

Table 4.1: Distribution of what the most similar counterpart a sound has within each class.

		Highest Similarity Score				
		1	2	3	4	5
Sound Class	Can opening	0	1	19	19	0
	Church bells	0	0	19	19	1
	Dog	0	1	19	17	2
	Pouring water	0	3	24	12	0
	Thunderstorm	0	3	17	17	2

Table 4.2 shows the average distribution of scores per class. It shows that samples have few similar sounds, with most having a score of 1 or 2, and a marginal number having a score of 3 or more.

Table 4.2: Average distribution of samples each scores has per class.

		Average Number of Samples With Each Score				
		1	2	3	4	5
Sound Class	Can opening	13.65	11.62	5.12	0.8	0.0
	Church bells	15.15	10.87	4.45	0.7	0.025
	Dog	15.525	11.55	3.57	0.5	0
	Pouring water	17.22	10.45	3.12	0.4	0
	Thunderstorm	16.27	9.7	4	1.15	0.025

4.4.1 Experiment – Random Sampling

As can be seen from Table 4.2, it is not easy to calculate what a randomly sampled accuracy would be. The easiest would be giving each score an equal probability of being pulled, allowing the use of simple fractions like $\frac{1}{50} = 2\%$. But the probability of polling a sample with a score of one or two is significantly higher than the other scores, and the opposite is true for four and five. Finding the random accuracy is then either done by more complex math or experimentation.

We chose experimentation as it is the easiest method. We designed two experiments where every audio file (query) in our data set was randomly assigned five different audio files from one of two data sets. The design of the experiments reflects the experiments performed in section 5.5 to allow comparison between the results.

The first experiment assigned sounds from the same class as the query. The second experiment assigned sounds from the entire ESC-50 data set. The list was ordered by when they were assigned. The top-1 accuracy was when the first assigned was one of the query's most similar samples (had the highest score). The top-5 was when either of the five sounds were one of the query's most similar samples. Both experiments were repeated 50 times, and the average is reported in Table 4.3.

Table 4.3: Accuracy when randomly drawing samples, trying to find the most similar sound.

Experiment	**Top-1 Accuracy**	**Top-5 Accuracy**
Experiment 1	9.4%	35.7%
Experiment 2	<1%	<1%

We see that randomly drawing the right answer from the same class is unlikely, but polling from the entire ESC-50 dataset is very unlikely.

Chapter 5

Study 2: Sound2Vec

We base our model on the ResNet models provided by PyTorch and perform transfer learning. PyTorch's ResNet models are pre-trained on ImageNet, following the same procedure as the designers of ResNet [64].

Below we explain how the models have been modified and retrained to extract fingerprints from spectrograms. We then present the experiments performed to test its capability to measure similarity.

5.1 Modifying ResNet

The ResNet models provided by PyTorch are designed for classifying ImageNet. That means it expects images as input and outputs 1000 probabilities to indicate the predicted class. To do feature extraction, the output layers must be replaced so that it outputs a desired number of features that are not distributed as probabilities. Referring to Figure 2.5, that means the '1000-d fc' and 'softmax' layers must be replaced. Also, the input layer (conv1) must be modified because spectrograms are not like color images.

The input layer must be modified because the images in ImageNet have three "channels", representing the red, green, and blue color values (see Figure 5.2). Computationally, these channels are three layers of data per image. A spectrogram only has a single layer.

The input layer is therefore modified to accept a single channel. The layer is modified because replacing it with an uninitialized

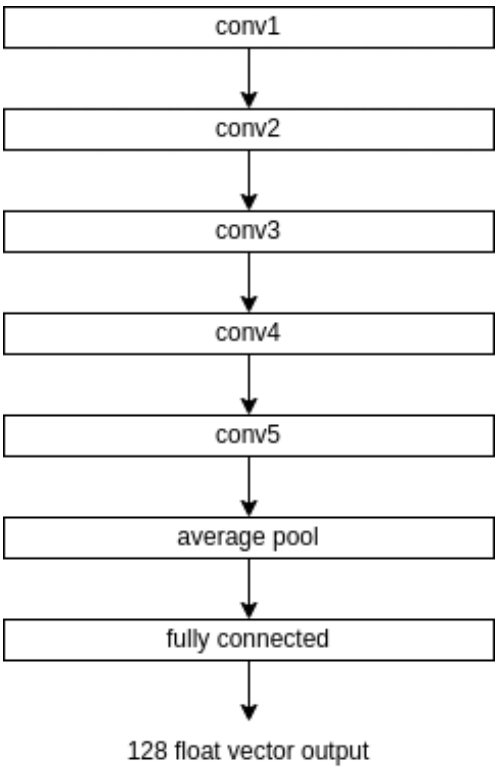


Figure 5.1: Final model architecture. It is the same for ResNet18 and 34. See Figure 2.5 for details about each layer.

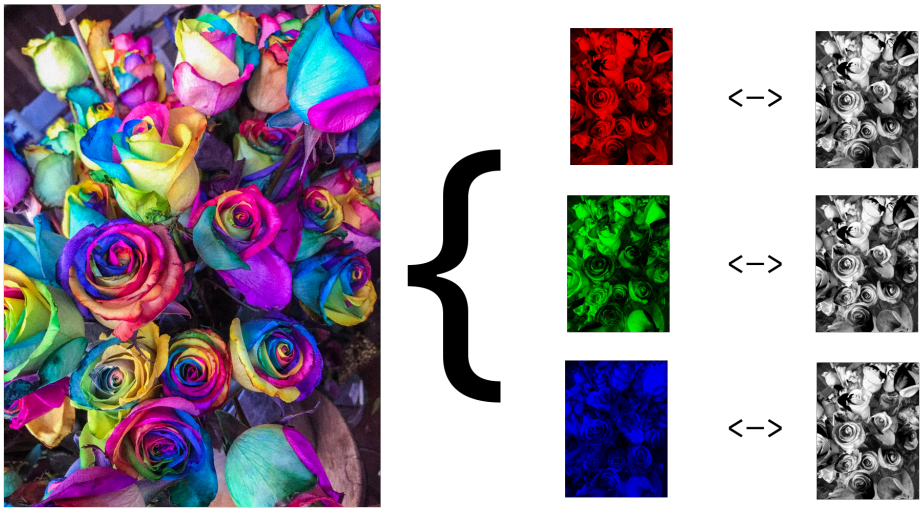


Figure 5.2: An image decomposed into its component RGB values (each image is really a gray-scale image (1 channel). The middle column is a stylization and the last column is what it actually looks like). Source: [66]

layer would invalidate the rest of the network, making transfer learning impossible. We therefore transfer the old layer's weights by taking the sum of the three channels. This method of merging the channels is taken from fastai [65][1].

The '1000-d fc' layer is replaced to change the number of output neurons to 128. Experimentally, we see that using 128 features increases accuracy, and more does not improve our evaluation metrics. We also remove the softmax layer because it converts the output to probabilities. See Figure 5.1 for an overview of the architecture of Sound2vec.

5.2 Spectrogram Processing

The literature establishes that image classification can be transferred to audio classification with remarkable success [35]. However, raw audio is represented as a one-dimensional signal. That means raw audio is unusable with 2-D Convolution neural networks, the backbone of image classification.

We therefore convert the audio into two-dimensional, image-like mel spectrograms. Spectrograms represent the signal in time and frequency and can efficiently be processed by convolutional neural networks.

Extracting spectrograms is a three-step process. First the audio waveforms

[1] https://github.com/fastai/fastai/blob/master/fastai/vision/learner.py

are normalized. Then the audio is converted into mel spectrograms, before being normalized again. The two following subsections describe the details of how the mel spectrograms are created as well as how the normalization is performed.

5.2.1 Waveform normalization

The first step is to normalize all the raw waveforms to ensure their uniformity. They are normalized by giving them a mean of zero and dividing it with their standard deviation. See Equation 5.1.

$$norm\_audio = \frac{audio - mean(audio)}{std(audio)} \tag{5.1}$$

5.2.2 Mel Spectrogram Parameters

The second step is to convert the normalized waveforms into mel spectrograms. We use the `Fastaudio` python library for this purpose [67]. It speeds up loading and conversion of the audio significantly compared to other audio libraries such as `librosa` [68].

`Fastaudio` requires several parameters from the user that both control the format the audio is converted into, as well as how it is converted into mel spectrograms. What they are can dramatically impact model performance and is an ongoing field of research [42][69]. We decided to use parameters that give high classification accuracy with ESC-50.

More specifically, we adopt optimal parameters decided by comprehensive fine-tuning [53]. The fine-tuning was performed with `FastAI` [65] and `Fastaudio`, and multiple values for each parameter were tested. See Table 5.1 for the parameters we used.

5.2.3 Spectrogram Normalization

Transfer learning often requires that the same normalization statistics are used. Considering vision-oriented uses of convolutional neural network use RGB images, the statistics are often tied to normal images. In the case of ResNet the statistics come from ImageNet.

However, spectrograms and images are on completely different scales. An image encodes each pixel with a value between 0 to 255 per channel. Each channel value can therefore easily be mapped between 0 and 1, and then normalized from there. Spectrograms, by contrast, can have values from $-\infty$ to $+\infty$. Spectrograms should therefore be normalized differently from images.

But changing the normalization statistics when using transfer learning is considered bad practice. The alternative is to train the model from scratch.

However, research shows that normalizing with statistics from ESC-50 is a boon; Even with a pre-trained ResNet model [53]. With the right spectrogram

Table 5.1: Parameters passed to spectrogram function

Full name	**Value**	**Parameter name**	**Description**
Sample Rate	44100 samples	sr	How often the analogue signal is sampled per second. Signals with too few/many samples are converted to the correct sample rate.
Number of FFT bins	4096 bins	n_fft	How many FFT bins to generate. Higher values can give better scores, but diminishing returns after 4096 bins.
Number of Mel Bands	224 mels	n_mels	How many mel-spaced bands to generate. Corresponds to the number of values on the X-axis.
Hop Length	308 samples	hop_length	How far the Fourier frame should shift to the right (from the center of the current frame). The frames will overlap if the value is less than 'n_fft'. Note that the size of the spectrogram is directly affected by this parameter. Halving the number of samples doubles the size, requiring more GPU memory.
Window Length	2205 samples	win_length	How many samples are included in each Fourier frame. Higher values lead to higher frequency resolution, but lower time resolution.
Frequency range	0 Hz to 18000 Hz	f_min to f_max	The range of frequencies the FFT bands will be split between. f_max defaults to the Nyquist frequency (sample rate/2). f_min defaults to 0.
Window Function	Hann window	window_fn	The windowing function used to window the signal.

parameters, it is possible to achieve a classification accuracy of up to 88%, beating both other transfer learning models and models trained from scratch.

We therefore normalize the mel spectrograms around ESC-50's statistics [54]. Using this technique, we find the statistics by averaging the mean and standard deviation of all spectrograms in the dataset. This gives a mean of -43.1299 and standard deviation of 27.4627.

Note that normalization is applied "globally" as opposed to "frequency-bin-based". "Global" means the normalization statistic is collected from and applied to every frequency-bin indiscriminately. Frequency-bin-based normalization might be necessary due to a potential significant difference between the bins. However, research indicates that global normalization is optimal for ESC-50 [54].

5.3 Training the Model

The model is trained to minimize distance between features from samples of the same class. The similarity is limited to a class-level because the annotated data set from study 1 is too small to train the model in a meaningful way.

Both ResNet18 and ResNet34 were trained using the hyperparameters presented below. The training is limited to 40 epochs because more epochs were shown to give negligible improvement in performance.

5.3.1 Loss Function

A loss function is some mathematical function that calculates the distance between a model's output and what was expected. Larger loss means the weights of the model change more. In this case, the loss function should influence the weights to maximize closeness with similar sounds and maximize distance to dissimilar sounds.

Triplet Margin Loss does exactly that. More specifically, it encourages similar pairs to be closer than dissimilar pairs by some margin. A similar "pair" being two samples with the same label.

Mathematically, the loss is calculated as $L = max(d(p,a) - d(n,a) + m, 0)$, where

- `d()` is some distance function
- `a` is the "anchor" – the sample we compare the other samples against
- `p` is a positive sample with the same label as `a`
- `n` is a negative sample with a different label from `a`
- `m` is the margin

In this thesis we use PyTorch's default `TripletMarginLoss` implementation, which is implemented following the research paper by Balntas et al. [70]. By default, it uses the euclidean distance as the distance function and 1.0 as the margin. The only alteration was to enable the "swap" parameter, which tells it to use $p-n$ instead of $a-n$, if it violates the margin more.

5.3.2 Optimizer

To train the network we use the Adam optimizer from pytorch. But instead of using a static learning rate, we implement the concept of cyclic learning. Cyclic learning is an alternative to instruct the optimizer to ignore certain layers during training, which is referred to "freezing" the layers.

Freezing the original network is common in transfer learning. With the proposed architecture, the model's ability to recognize shapes and objects would be retained, and only a "post-processing" layer would be trained. In general, it leads to requiring a smaller data set and less training time.

However, if the input is abnormal, which is the case of spectrograms, the model's understanding might not transfer well.

Hartquist showed that it is beneficial to freeze the network only partially [53]. In his work, he used the ResNet18 network from FastAI; which only freezes the convolutional layers. In an experiment, we compared his network with a completely frozen network. The results show that the partially frozen FastAI network has a 10-percentage point benefit to classification accuracy compared to a completely frozen network.

Knowing that training the entire model was beneficial, we used the technique proposed by Mushtaq et al. to train our model [59]. They kept the entire network unfrozen and trained it in using different learning rates for certain layers; increasing model performance over the "normal" method.
We replicate the learning methodology that they used. They divided the network into three groups that were given their own learning rate [59]. The initial group are the top-most layers. It determines simple structures like lines in the image. This layer is useful for almost any visual task and has a low learning rate. The next group is the middle layers. They determine patterns like rectangles, squares, etc. and have a higher learning rate. The last layers detect more complex patterns and are given the highest learning rate.

We employ learning rates of 1e-5, 1e-4, and 1e-3. See Table 5.2.

Table 5.2: The cyclic learning rates for ESC-50

LR	**Layers**
1e-5	conv1, conv2, conv3
1e-4	conv4, conv5
1e-3	fc

5.3.3 Scheduler

Keeping the learning rate static throughout increases the time it takes to reduce loss. The model would get "stuck" on a loss value for a longer period before decreasing. We therefore introduce a scheduler to reduce the learning rate during training and speed up convergence.

We use the Multi Step Learning Rate scheduler from `PyTorch`. It reduces the learning rate by a factor of `Gamma` when it reaches a `Milestone` epoch. This increases model performance by 0-4 percentage points compared to a model without a scheduler.

Milestones	**Gamma**
10, 20, 30	1/10

5.3.4 Hardware and Software

The model was implemented using the software and hardware listed in Table 5.3.

Table 5.3: Hardware and Software used during the experiments.

Hardware/Software	**Purpose**
Fedora 35	Operating system
GTX 1070 8GB	Compute
32 GB RAM	Random Access Memory
PyTorch	Machine Learning Framework
Fastaudio	Loading and processing audio for use with PyTorch.
Pandas	Processing of data set CSV files
Crucial CT500MX	Storage. Affects the speed of loading sound files from disk.

5.4 Finding Matches

A method to search through a database is necessary to find similar sounds. This involves linearly comparing distances between a query fingerprint and a database of fingerprints.

The simplest matching methodology is to compare the query against every single entry in the database. This kind of search ensures the best match in the database, but the search complexity grows with the database ($O(n)$).

An alternative is Locality Sensitive Hashing (LSH). LSH also employs linear search, but on a smaller scale. It hashes similar input into the same buckets, and only needs to search through the one bucket a query is hashed into. The search complexity is therefore reduced to $O(n/number\_of\_buckets)$. All queries to LSH must be in the same format as all the data that is already stored. It differs from normal hashing by using hashes that maximize collisions.

Whenever a query is made, the query is linearly matched against entries in the bucket it was hashed into. Because it matches every sample in a bucket, it can return a list of ‘n‘ samples ordered after distance from the query.

However, by only searching in one bucket, it cannot guarantee the globally most similar match. We can mitigate this by using multiple hash tables to look up multiple indexes at once. This can increase the probability of finding the global minimum at the cost of some extra computations.

To implement LSH we use the python library "LSHashPy3".[2] It allows us to store a large database in memory and query for the ‘n‘ closest samples. It calculates hashes by converting input into bit-strings:

1. Generating a random array at startup. The length is user specified.
2. Performing a dot-product between the input and the randomly generated float. The product is the length of the random array.
3. Binary-stepping the product, where values greater than zero are ’1’, and everything else is ’0’. The bit-string is the hash.

[2]https://pypi.org/project/lshashpy3/

LSHash also supports multiple hash tables with different hash-arrays. This allows a single search to look through multiple hash-indexes.

For this thesis we use a hash length of 1, and 5 hash tables. This way we increase reproducibility, as LSHash does not support seeding the random number generator.

5.5 Experimental Procedure

Four experiments were carried out to test the model's performance. Experiments one and four are classification experiments, while experiments two and three focus on similarity. This introductory section explains the procedure of the experiments, while the following subsections present the results of the experiments.

Each experiment was performed once using ResNet18 and once using ResNet34 as the base in Sound2Vec. Every time the top-1 accuracy, top-5 accuracy, and the average time taken were recorded. In the following we present the general experimental procedure, and then the specifics of each experiment.

The general procedure:

1. Convert the data set into our extracted 128-dimensional features using Sound2Vec.
2. Using the pre-defined cross-validation folds: split the data set into testing and training, where the training data set is the lookup database.
3. For every entry in the testing data set, query the database and receive a list of the five most similar sounds.
4. (experiment 2 and 3) Look up the highest possible similarity score available to each entry.
5. Register Top-1 and Top-5 accuracy, based on if either of them contains the highest possible similarity score or the correct class.

All the experiments follow the K-fold cross-validation prearranged from ESC-50 and UrbanSound8K. That means the data set was split into five and ten folds, respectively.

Experiment 1 tests the model's ability to create similar features for sounds of the same class. It classifies the ESC-50 data set. An input sound is converted to 128 features and used for classification by searching for similar sounds in the lookup database. Each entry in the lookup database includes the class. The sound is correctly classified if the most similar sound has the same class as the query. This metric enables comparisons with other audio classification algorithms.

Experiment 2 tests the features' ability to differentiate between sounds of the same class. That is, the features from a sound are compared to the features of other sounds from the same class. The comparison returns five ranked sounds that are predicted to be similar, where the highest ranked is the most similar. It is

"correct" if the predicted most similar sound is the most similar sound according to our data set.

Experiment 3 combines experiment one and two and tests the features' ability to first classify and then find similar samples. Only the sounds in our data set are classified, but it uses the entire ESC-50 data set as the lookup database. Classification is performed as in experiment 1 and if the class is correct, it also performs similarity measuring like in experiment 2. If the class is wrong, then it is recorded in the results as getting it wrong. The goal is to see how this differs from the accuracy reported by experiment 2. If the difference is negligible, then the sounds that are correctly classified are more likely to be "clearly" similar.

Experiment 4 classifies the UrbanSound8K data set using models that are trained on ESC-50. In other words, it is not trained on UrbanSound8K. This is to show if the model understands of similarity between sounds or not.

5.5.1 Experiment 1 Results – Classification

The first experiment considers if the top-1 or top-5 contains the same class as the query. The experiment is carried out using both ESC-50 and ESC-10.

ESC-50

Table 5.4 shows the experimental results for classifying ESC-50. The results show average accuracy and time used per fold. Each fold has 400 samples to classify and a database of 1600 samples. To reiterate, this data set is more complex than ESC-10 and should be more difficult to classify.

Table 5.4: Sound2Vec's classification results when evaluated ESC-50.

Base Model	Top-1 Accuracy	Top-5 Accuracy	Lookup Time	Lookup Time per sound
ResNet18	73.2%	86.6%	20 seconds	0.05 seconds
ResNet34	74.9%	88.3%	25 seconds	0.0625 seconds

ESC-10

Table 5.5 shows the classification results for ESC-10. The results show average accuracy and time used per fold. Each fold has 80 samples to classify and a database of 400 samples.

Table 5.5: Sound2Vec's classification results when evaluated ESC-10.

Base Model	Top-1 Accuracy	Top-5 Accuracy	Lookup Time	Lookup Time per sound
ResNet18	89.5%	95.25%	3.6 seconds	0.045 seconds
ResNet34	86.25%	91.25%	4.7 seconds	0.05875 seconds

5.5.2 Experiment 2 Results – Intra-Class Similarity

This experiment considers if the top-1 and top-5 contain the most similar sound according to our data set. The experiment is performed once for every class in our similarity-data set, with the class being the entire data set. The average accuracy and time were recorded in the results in Table 5.6. Each class had 8 sounds to find similar sounds for, and a database of 40 samples.

Table 5.6: Average accuracy when finding similar sounds in a class (Intra-Class similarity).

Base Model	Top-1 Accuracy	Top-5 Accuracy	Lookup Time	Lookup Time per sound
ResNet18	27%	51%	6 seconds	0.15 seconds
ResNet34	26%	61%	6.3 seconds	0.1575 seconds

5.5.3 Experiment 3 Results – Inter-Class Similarity

The third experiment classifies the sound before finding similar sounds. The average accuracy and results are presented in Table 5.7. Each class had 8 sounds to find similar sounds to, and a database of 1600 samples.

Table 5.7: Average accuracy when classifying the sound and then finding similar sounds within the class (Inter-Class Similarity).

Base Model	Top-1 Accuracy	Top-5 Accuracy	Lookup Time	Lookup Time per sound
ResNet18	26%	51.5%	130 seconds	3.25 seconds
ResNet34	25.5%	55.55%	160 seconds	4 seconds

As we can see, doing both classification *and* similarity matching impacts the speed of the algorithm significantly.

5.5.4 Results Experiment 4 – Classifying UrbanSound8K

The last experiment classifies UrbanSound8K without training Sound2Vec on it. The average accuracy and time are presented in Table 5.8. Each fold had 873 sounds to classify, and a database of 7858 samples.

The accuracy and speed are expected to be lower than classifying ESC-50. UrbanSound8K has more than four times as many sounds as ESC-50, and only ten classes with similar sounds.

Table 5.8: Accuracy when using Sound2Vec, trained on ESC-50, to classify UrbanSound8K.

Base Model	**Top-1 Accuracy**	**Top-5 Accuracy**	**Lookup Time**	**Lookup Time per sound**
ResNet18	63.1 %	83.7%	165 seconds	0.19 seconds
ResNet34	64.4%	83.8%	170 seconds	0.195 seconds

Chapter 6

Discussion

The focus of this thesis has been to investigate if sounds can be compared quantitatively using fingerprinting/feature extraction. The thesis also tested if image classifiers can be used for this purpose, and how well the features can be used to classify sounds or find similar sounds. The goal has been to develop a feature extraction machine learning framework based on ResNet, as well as to create a data set to verify the framework's abilities.

In this chapter we discuss the results from study 1 and 2 and their limitations, as well as the applicability of our results to the field of forensics.

6.1 Data Set Statistics

We see that it is common for sounds in the data set to be significantly similar to at least one other sound. In general, between 40-50% of the samples have a corresponding sound that has a similarity score of 4 or more. However, this number should be higher on a data set used for similarity. A sound will only have one to three sounds that have a similarity score of 4, but five to fifteen with a score of 3! Meaning that the probability of finding the "most similar" is much higher for 3s than for 4s. This most likely inflates the accuracy significantly.

For this specific data set, it is expected that many of the sounds would have lower scores. The developer of the ESC-50 data set explicitly noted that the sounds were chosen because they showed a broad range of the sound [38]. This is perfect for learning classification but makes it more difficult to pinpoint the most similar sounds.

6.2 Discussion of Classification Results

The results of the classification experiment indicate that the extracted features can be used to classify sounds (Table 5.4). The experiment achieves a top-1 accuracy between 73% to 75% on the ESC-50 data set, a score that is notably higher

than the established baseline for CNNs at 64.5% [48]. This shows that the features can be discriminated against on a class-level. In other words, we can extract fingerprints that can be used to quantitatively compare causal similarity.

But these results are hardly state-of-the-art. Even compared to other ResNet models it performs merely adequately. It matches a ResNet18 network trained from scratch at 73.15% [54] and falls significantly behind other models with the same preconditions, at 89.54% [53]. That said, the lower accuracy might be due to ambiguity introduced by the data set, where recordings sound similar even though they are of different classes. For example, the sound of rain and sound of waves can be remarkably similar. This can be seen in the top-5 accuracy, where the right class is among the candidates 86-88% of the time. More samples and a broader range of recordings per class could increase top-1 accuracy.

The results also show that the depth of the underlying network has negligible impact on accuracy. ResNet18 and 34 trade blows, but neither beats the other by more than 2 percentage points. ResNet18 barely scores higher on the significantly smaller ESC-10 data set but is beat by ResNet34 on ESC-50. This is in line with previous research, where the more complex ResNet networks will perform similarly to ResNet18 [53].

The experiments also show that the classification is fast. The time required to classify 80 sounds is about 20-25 seconds, meaning that a single sound can be classified in around 0.25-0.3 seconds using a database with 1600 samples. This should be more than fast enough to annotate sounds in real time, considering each recording is 5 seconds long. This speed can be retained with larger data sets if a database searching method like Locality Sensitive Hashing is used.

The results from classifying UrbanSound8K (Table 5.8) shows that Sound2Vec has inclinations towards a general understanding of sounds. The model was not trained on the data set but was still able to achieve around 64% accuracy. While not state-of-the-art, it shows one of the benefits of using a fingerprinting technique instead of classification: the model can classify new classes without retraining. Though, the similarity is most likely wholly based on acoustical similarity: Since the model has never seen the sounds before, it relies heavily on very similar sounds to be available in the database to correctly classify it, which is likely with such a large data set with few classes. Achieving this accuracy is unlikely on a data set with many hard to distinguish classes.

6.3 Discussion of Similarity Results

The results after the intra-class experiments (in Table 5.6) and Table 5.7) shows a non-random top-1 accuracy of 26-27% compared to a random accuracy of 9.4% (Table 4.3). This accuracy is not ground-breaking, but still significant. This shows that the features extracted by Sound2Vec can differentiate between semantically similar sounds, but not at a significant level.

Though, because of the subjective nature of the data set, the top-1 accuracy might not be a good metric. It is not guaranteed that what the data set states is the

most similar sound, is the most similar sound. There might be more similar sounds that the annotator ignored, but the model picked up on. A qualitative review of the answers from the model shows that the sounds it puts as top-1 are often very similar to the query.

It is therefore more reasonable to focus on the top-5 accuracy. We see an accuracy of 51-61% against 35.7% random accuracy. This difference is significantly higher than the top-1 accuracy, with a percentage point increase larger than the top-1 vs top-5 accuracy of the classification experiments. The reason for achieving a greater score here is probably because there are multiple chances that the most similar sound is drawn. Interestingly, ResNet34 outperforms ResNet18's Top-5 accuracy significantly in both experiments. This is probably down to ResNet34's deeper model, enabling more detail about the sounds to be extracted. A question then is if even deeper models would perform even better.

When considering inter-class similarity (Table 5.7, the accuracy stays more or less similar to the intra-class similarity (Table 5.6). This is expected. The sounds it got right would have been clearly separable from others, meaning there is an overlap between the sounds it got right in intra-class similarity and classification. Since inter-class similarity is classification *and* similarity, the scores are similar.

But using the feature extraction mechanism for classification and similarity is not advisable. The recovery speed is slow. Though the similarity measure is slow per class, it achieves acceptable speeds when searching through the class only. A complete solution should therefore instead use an independent classification method and then apply the similarity measure on the relevant class.

Both the similarity and classification accuracy results show that the features can measure causal and semantic similarity. This then answers research questions 1 and 3. The fingerprints we extract can measure similarity between sound with the speed which can be described as fast if the data set is small. And the measure of similarity can be used to both classify and describe the sound. "Describing" being finding a similar sound in a database that has a description and using the same description on both.

Then, since the feature extraction model is based on image classifiers, we can answer research questions 2: Image classifiers *can* be used to extract similar features for similar sounds.

6.4 Limitations

The semantic similarity measuring was significantly impacted by the lack of a high-quality data set. The model could not be trained to differentiate between similar sounds, and the data set was too biased to say anything about how general the results are.

There are multiple limitations with the methodology that speak against the generalization of the data set. The most impactful is the small pool of participants.

By only having a single annotator, the data set is heavily biased towards their perception of similarity. But even their understanding of similarity is probably skewed. The general limitations of pairwise comparisons would have affected the listener's ability to accurately annotate similarity. Also, the restrictive nature of using a scale of 1 to 5 would affect the quality, as it is difficult to apply uniformly.

That said, it is very unlikely that the annotations are very wrong. All humans probably have a similar understanding of sound, and a listener should therefore annotate dissimilar sounds as dissimilar. The data set should therefore highlight which sounds are similar to some extent correctly, even if the actual scoring is suboptimal. Though, this is just speculation, and should be verified in the future.

6.5 Applicability to Forensics

Evidence brought into court should be as infallible as possible, which means that audio forensics evidence should be too [71]. The tooling must therefore be provably unbiased, have known reliability statistics, and be widely accepted by the forensics community [2]. Our models or data set do not meet these requirements, being dependent on subjective similarity and being new. This technique should therefore not be automatically applied to the transcription task without oversight, nor the sound identification task without verification.

Though, because of these limitations, many courts rely on human expert interpretation, and investigators only use tools to find sections in audio of interest or aid them in concluding on what a sound is [2]. In this situation, our model can be used to find specific-sounding sounds in the audio or used to provide a draft transcription of the audio that a human annotator can edit.

Chapter 7

Conclusion & Future Work

7.1 Conclusion

In this thesis we investigated if a feature extraction model can be used to compare sound similarity. To answer this, we developed Sound2Vec from the ResNet-family of image classifiers, as well as our own test data set that shows the subjective similarity between sounds from the ESC-50 data set [9]. Results show that features extracted using Sound2Vec can measure both causal and semantic similarity between sounds. Meaning, respectively, the features can be used for classification, as similar sounds will likely be the same class, and the features are descriptive enough to discriminate between similar and dissimilar sounds within the same class. Additionally, it can compare sounds very quickly depending on the size of the reference database.

Overall, the results in this thesis shows that image classifiers can be used for feature extraction to enable comparison of both causal and semantic similarity. Sound2Vec recognizes differences in sounds and can display them using discriminating features. Finding similar sounds then allows us to describe the sound on a class or deeper level.

We believe that the methodology proposed here could become a part of an audio forensics investigator's toolkit, aiding them in finding audio and annotating recordings.

7.2 Future Work

As part of this thesis, we proposed Sound2Vec – a feature extraction algorithm for sound similarity. While Sound2Vec works on "sterile" data sets, such as ESC-50, UrbanSound8K, and our own, it should be tested on more noisy and complex data sets. Additionally, it should be able to work on small, domain-specific data sets, as well as raw audio input to be used in the real world. This would notably require considering how it should split the audio stream into chunks and detect the presence of sounds.

The data set we created for this thesis is biased and small. Creating a larger, higher quality data set that considers bias should be done in future work. To do this, one could use a faster annotation method to reduce the workload. Reducing the workload would also make it easier to include more than one participant.

One way to annotate similarity could be using triplets. Triplet-annotation comes from the fact that listeners will describe an environmental sound by three attributes: the object making the noise; the action taken upon that object; and where the action took place [10]. For example, "A single wood plank dropped on concrete in a tunnel".

The benefit is that it enables comparing the similarity of sounds with only a single listening of each recording. And it can deliver this without introducing undue uncertainty in annotation-accuracy. Additionally, a certain level of absolute truth is available. There truly was an object, action, and location tied to that sound, and a data set where the audio is recorded by the researcher could have absolute certainty about these properties.

However, each attribute should be described with words or phrases from a (small) defined dictionary, and as such there is little room for overlap, ambiguity, or acoustic variations (pitch, timbre, etc.). Creating the dictionary and applying it correctly also requires deep knowledge of the sound in question. How to do this should be considered in future work.

Bibliography

[1] C. Williams, "Met lab claims 'biggest breakthrough since watergate'," *The Register*, Jan. 1, 2010. [Online]. Available: https://www.theregister.com/2010/06/01/enf_met_police/ (visited on 06/01/2022).

[2] R. C. Maher, "Audio forensic examination," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 84–94, 2009. DOI: 10.1109/MSP.2008.931080.

[3] S. Chachada and C.-C. J. Kuo, "Environmental sound recognition: A survey," *APSIPA Transactions on Signal and Information Processing*, vol. 3, e14, 2014. DOI: 10.1017/ATSIP.2014.12.

[4] S. Chandrakala and S. L. Jayalakshmi, "Environmental audio scene and sound event recognition for autonomous surveillance: A survey and comparative studies," *ACM Comput. Surv.*, vol. 52, no. 3, Jun. 2019, ISSN: 0360-0300. DOI: 10.1145/3322240. [Online]. Available: https://doi.org/10.1145/3322240.

[5] S. Raponi, I. Ali, and G. Oligeri, *Sound of guns: Digital forensics of gun audio samples meets artificial intelligence*, 2020. DOI: 10.48550/ARXIV.2004.07948. [Online]. Available: https://arxiv.org/abs/2004.07948.

[6] F. Beritelli and A. Spadaccini, "Human identity verification based on mel frequency analysis of digital heart sounds," in *2009 16th International Conference on Digital Signal Processing*, 2009, pp. 1–5. DOI: 10.1109/ICDSP.2009.5201109.

[7] P. X. Zhang, "Chapter 3 - psychoacoustics," in *Handbook for Sound Engineers (Fourth Edition)*, G. M. Ballou, Ed., Fourth Edition, Oxford: Focal Press, 2008, pp. 41–63, ISBN: 978-0-240-80969-4. DOI: https://doi.org/10.1016/B978-0-240-80969-4.50007-9. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780240809694500079.

[8] K. J. Piczak, *Esc-50: Dataset for environmental sound classification*, https://github.com/karolpiczak/ESC-50, 2015.

[9] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*, Brisbane, Australia: ACM Press, Oct. 13, 2015, pp. 1015–1018, ISBN: 978-1-4503-3459-4. DOI: 10.1145/2733373.2806390. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2733373.2806390.

[10] N. VanDerveer, "Ecological acoustics: Human perception of environmental sounds," Ph.D. dissertation, 1979. [Online]. Available: https://www.proquest.com/openview/d2ccf8de5da2c0eea3e78d631a2bb087/1?pq-origsite=gscholar&cbl=18750&diss=y.

[11] W. Gaver, "How do we hear in the world? explorations in ecological acoustics," *Ecological Psychology*, vol. 5, pp. 285–313, Dec. 1993. DOI: 10.1207/s15326969eco0504_2.

[12] W. Gaver, "What in the world do we hear?: An ecological approach to auditory event perception," *Ecological Psychology*, vol. 5, pp. 1–29, Mar. 1993. DOI: 10.1207/s15326969eco0501_1.

[13] G. Lemaitre, O. Houix, N. Misdariis, and P. Susini, "Listener expertise and sound identification influence the categorization of environmental sounds," *Journal of experimental psychology. Applied*, vol. 16, pp. 16–32, Mar. 2010. DOI: 10.1037/a0018762.

[14] J. A. Ballas and T. Mullins, "Effects of context on the identification of everyday sounds," *Human Performance*, vol. 4, pp. 199–219, 1991.

[15] J. Howard and J. Ballas, "Syntactic and semantic factors in the classification of nonspeech transient patterns," *Perception psychophysics*, vol. 28, pp. 431–9, Dec. 1980. DOI: 10.3758/BF03204887.

[16] J. Ballas, "Common factors in the identification of an assortment of brief everyday sounds," *Journal of experimental psychology. Human perception and performance*, vol. 19, pp. 250–67, May 1993. DOI: 10.1037/0096-1523.19.2.250.

[17] J. Ballas and J. Howard, "Interpreting the language of environmental sounds," *Environment and Behavior*, vol. 19, pp. 91–114, Jan. 1987. DOI: 10.1177/0013916587191005.

[18] K. M. Aldrich, E. J. Hellier, and J. Edworthy, "What determines auditory similarity? the effect of stimulus group and methodology," *Quarterly Journal of Experimental Psychology*, vol. 62, no. 1, pp. 63–83, 2009, PMID: 18609397. DOI: 10.1080/17470210701814451. eprint: https://doi.org/10.1080/17470210701814451. [Online]. Available: https://doi.org/10.1080/17470210701814451.

[19] K. Aldrich, E. Hellier, and J. Edworthy, "What determines auditory similarity? the effect of stimulus group and methodology," *Quarterly Journal of Experimental Psychology*, vol. 62, pp. 63–83, 2009.

[20] G. Scavone, S. Lakatos, P. Cook, and C. Harbke, "Perceptual spaces for sound effects obtained with an interactive similarity rating program," Sep. 2001.

[21] A. Guillaume, L. Pellieux, V. Chastres, and C. Drake, "Judging the urgency of nonvocal auditory warning signals: Perceptual and cognitive processes," *Journal of experimental psychology. Applied*, vol. 9, pp. 196–212, Sep. 2003. DOI: 10.1037/1076-898X.9.3.196.

[22] G. Lemaitre, O. Houix, N. Misdariis, and P. Susini, "Listener expertise and sound identification influence the categorization of environmental sounds.," *Journal of experimental psychology. Applied*, vol. 16 1, pp. 16–32, 2010.

[23] A. C. Guyton and J. E. Hall, *Textbook of medical physiology*, en, 11th ed., ser. Guyton Physiology. London, England: W B Saunders, Jul. 2005.

[24] K. Steiglitz, *A digital signal processing primer: With applications to digital audio and computer music A digital signal processing primer: With applications to digital audio and computer music*. Mineola, NY: Dover Publications, Nov. 2020.

[25] M. H. Weik, *Communications standard dictionary on CD-ROM*, en, 3rd ed. London, England: Chapman and Hall, Dec. 1995.

[26] C. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949. DOI: 10.1109/JRPROC.1949.232969.

[27] F. Harris, *Multirate Signal Processing for Communication Systems*. Prentice Hall PTR, 2004, ISBN: 9780131465114. [Online]. Available: https://books.google.no/books?id=ve5SAAAAMAAJ.

[28] J.-B.-J. Fourier, *Théorie analytique de la chaleur*. F. Didot, 1822, ISBN: 9780511693229. DOI: https://doi.org/10.1017/CBO9780511693229.

[29] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of Computation*, vol. 19, pp. 297–301, 1965.

[30] N. Project, *Discrete fourier transform*, https://numpy.org/doc/stable/reference/routines.fft.html#module-numpy.fft.

[31] F. A. R. lab (FAIR), *Torch.stft*, https://pytorch.org/docs/stable/generated/torch.stft.html.

[32] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The Journal of the Acoustical Society of America*, vol. 8, no. 3, pp. 185–190, 1937. DOI: 10.1121/1.1915893. eprint: https://doi.org/10.1121/1.1915893. [Online]. Available: https://doi.org/10.1121/1.1915893.

[33] D. D. Greenwood, *Auditory - research in auditory perception*, https://web.archive.org/web/20130208164732/http://lists.mcgill.ca/scripts/wa.exe?A2=ind0907d&L=auditory&P=389, 2009.

[34] Krishna Vedala, *File:mel-hz plot.svg*, [Online; accessed May 09, 2022], 2013. [Online]. Available: https://commons.wikimedia.org/wiki/File:Mel-Hz_plot.svg.

[35] A. Guzhov, F. Raue, J. Hees, and A. Dengel, *Esresnet: Environmental sound classification based on visual domain models*, 2020. DOI: 10.48550/ARXIV.2004.07301. [Online]. Available: https://arxiv.org/abs/2004.07301.

[36] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *22nd ACM International Conference on Multimedia (ACM-MM'14)*, Orlando, FL, USA, Nov. 2014, pp. 1041–1044.

[37] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[38] Y. Gong, Y. Chung, and J. R. Glass, "AST: audio spectrogram transformer," *CoRR*, vol. abs/2104.01778, 2021. arXiv: 2104.01778. [Online]. Available: https://arxiv.org/abs/2104.01778.

[39] Y. Su, K. Zhang, J. Wang, and K. Madani, "Environment sound classification using a two-stream cnn based on decision-level fusion," *Sensors*, vol. 19, no. 7, p. 1733, Apr. 2019, ISSN: 1424-8220. DOI: 10.3390/s19071733. [Online]. Available: http://dx.doi.org/10.3390/s19071733.

[40] H. Wang, Y. Zou, D. Chong, and W. Wang, *Environmental sound classification with parallel temporal-spectral attention*, 2019. DOI: 10.48550/ARXIV.1912.06808. [Online]. Available: https://arxiv.org/abs/1912.06808.

[41] Z. Zhang, S. Xu, S. Cao, and S. Zhang, *Deep convolutional neural network with mixup for environmental sound classification*, 2018. arXiv: 1808.08405 [cs.SD].

[42] V. Boddapati, A. Petef, J. Rasmusson, and L. Lundberg, "Classifying environmental sounds using image recognition networks," *Procedia Computer Science*, vol. 112, pp. 2048–2056, 2017, Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 21st International Conference, KES-20176-8 September 2017, Marseille, France, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2017.08.250. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050917316599.

[43] X. Zhao and D. Wang, "Analyzing noise robustness of mfcc and gfcc features in speaker identification," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 7204–7208. DOI: 10.1109/ICASSP.2013.6639061.

[44] J.-C. Wang, J.-F. Wang, K. W. He, and C.-S. Hsu, "Environmental sound classification using hybrid svm/knn classifier and mpeg-7 audio low-level descriptor," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, 2006, pp. 1731–1735. DOI: 10.1109/IJCNN.2006.246644.

[45] G. Muhammad, Y. A. Alotaibi, M. Alsulaiman, and M. N. Huda, "Environment recognition using selected mpeg-7 audio features and mel-frequency cepstral coefficients," in *2010 Fifth International Conference on Digital Telecommunications*, 2010, pp. 11–16. DOI: 10.1109/ICDT.2010.10.

[46] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015. DOI: 10.1109/MSP.2014.2326181.

[47] L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014, ISSN: 1932-8346. DOI: 10.1561/2000000039. [Online]. Available: http://dx.doi.org/10.1561/2000000039.

[48] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2015, pp. 1–6. DOI: 10.1109/MLSP.2015.7324337.

[49] B. Zhu, C. Wang, F. Liu, J. Lei, Z. Lu, and Y. Peng, *Learning environmental sounds with multi-scale convolutional neural network*, 2018. DOI: 10.48550/ARXIV.1803.10219. [Online]. Available: https://arxiv.org/abs/1803.10219.

[50] D. Plakal Manoj; Ellis, *Yamnet*, https://tfhub.dev/google/yamnet/1.

[51] A. Kumar, M. Khadkevich, and C. Fugen, *Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes*, 2017. DOI: 10.48550/ARXIV.1711.01369. [Online]. Available: https://arxiv.org/abs/1711.01369.

[52] X. Jin, Y. Yang, N. Xu, J. Yang, J. Feng, and S. Yan, *WSNet: Learning compact and efficient networks with weight sampling*, 2018. [Online]. Available: https://openreview.net/forum?id=H1I3M7Z0b.

[53] J. Hartquist, *Fine-tuning resnet-18 for audio classification*, Oct. 2020. [Online]. Available: https://wandb.ai/jhartquist/fastaudio-esc-50/reports/Fine-tuning-ResNet-18-for-Audio-Classification--VmlldzoyNjU3OTQ.

[54] C. Kroenke, *How to normalize spectrograms*, Sep. 2020. [Online]. Available: https://enzokro.dev/spectrogram_normalizations/2020/09/10/Normalizing-spectrograms-for-deep-learning.html.

[55] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015. DOI: 10.1007/s11263-015-0816-y.

[56] K. Palanisamy, D. Singhania, and A. Yao, *Rethinking cnn models for audio classification*, 2020. DOI: 10.48550/ARXIV.2007.11154. [Online]. Available: https://arxiv.org/abs/2007.11154.

[57] Y. Gong, Y.-A. Chung, and J. Glass, "Psla: Improving audio tagging with pretraining, sampling, labeling, and aggregation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021. DOI: 10.1109/TASLP.2021.3120633.

[58] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio Spectrogram Transformer," in *Proc. Interspeech 2021*, 2021, pp. 571–575. DOI: 10.21437/Interspeech.2021-698.

[59] Z. Mushtaq, S.-F. Su, and Q.-V. Tran, "Spectral images based environmental sound classification using cnn with meaningful data augmentation," *Applied Acoustics*, vol. 172, p. 107 581, 2021, ISSN: 0003-682X. DOI: https://doi.org/10.1016/j.apacoust.2020.107581. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0003682X2030685X.

[60] Coultas Blum, Harry A and Scart, Lucas G. and Bracco, Robert, *Introduction to audio for fastai students*, [Online; accessed May 09, 2022; MIT License], 2020. [Online]. Available: %5Curl%7Bhttps://fastaudio.github.io/Introduction%5C%20to%5C%20Audio%7D.

[61] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. DOI: 10.48550/ARXIV.1512.03385. [Online]. Available: https://arxiv.org/abs/1512.03385.

[62] Greg Goebel, *File:beer can pop-top display, budweiser brewery.jpg*, [Online; accessed May 18, 2022], 2014. [Online]. Available: https://commons.wikimedia.org/wiki/File:Beer_can_pop-top_display,_Budweiser_Brewery.jpg.

[63] P. Pongrácz, C. Molnár, and Á. Miklósi, "Barking in family dogs: An ethological approach," *The Veterinary Journal*, vol. 183, no. 2, pp. 141–147, 2010, ISSN: 1090-0233. DOI: https://doi.org/10.1016/j.tvjl.2008.12.010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1090023308004437.

[64] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. DOI: 10.48550/ARXIV.1512.03385. [Online]. Available: https://arxiv.org/abs/1512.03385.

[65] J. Howard and S. Gugger, "Fastai: A layered API for deep learning," *CoRR*, vol. abs/2002.04688, 2020. arXiv: 2002.04688. [Online]. Available: https://arxiv.org/abs/2002.04688.

[66] Uriel, *Closeup of multicolored petaled roses photo – free flower image on unsplash*, [Online; accessed May 16, 2022]. Published under the Unsplash License, 2022. [Online]. Available: https://unsplash.com/photos/WS4JcpoZz6E.

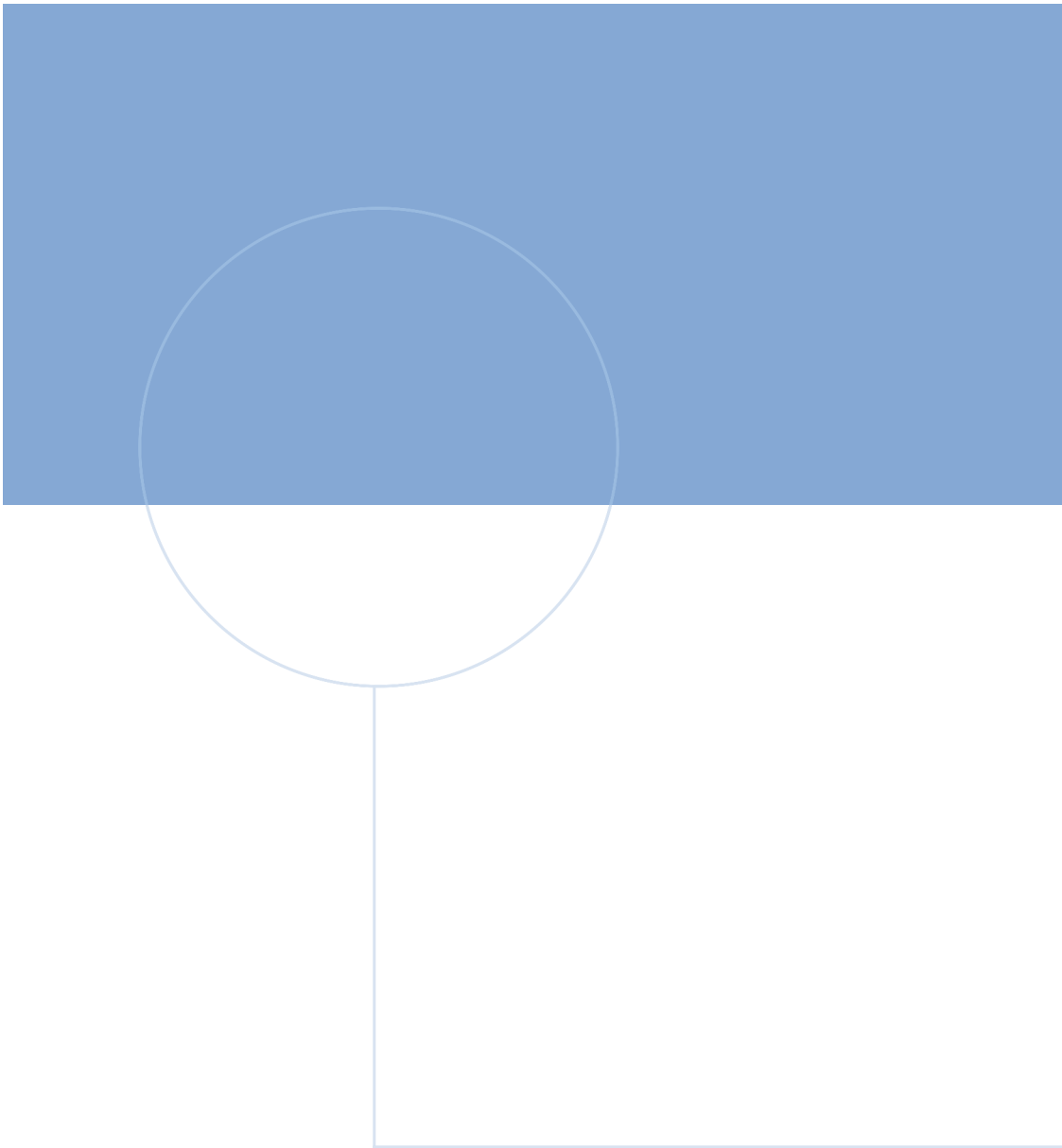
[67] H. A. Coultas Blum, L. G. Scart, and R. Bracco, *Fastaudio*, Aug. 2020. [Online]. Available: %5Curl%7Bhttps://github.com/fastaudio/fastaudio%7D.

[68] B. McFee, A. Metsai, M. McVicar, S. Balke, C. Thomé, C. Raffel, F. Zalkow, A. Malek, Dana, K. Lee, O. Nieto, D. Ellis, J. Mason, E. Battenberg, S. Seyfarth, R. Yamamoto, viktorandreevichmorozov, K. Choi, J. Moore, R. Bittner, S. Hidaka, Z. Wei, nullmightybofo, A. Weiss, D. Hereñú, F.-R. Stöter, P. Friesch, M. Vollrath, T. Kim, and Thassilo, *Librosa/librosa: 0.9.1*, version 0.9.1, Feb. 2022. DOI: 10.5281/zenodo.6097378. [Online]. Available: https://doi.org/10.5281/zenodo.6097378.

[69] F. Demir, D. A. Abdullah, and A. Sengur, "A new deep cnn model for environmental sound classification," *IEEE Access*, vol. 8, pp. 66 529–66 537, 2020. DOI: 10.1109/ACCESS.2020.2984903.

[70] D. P. Vassileios Balntas Edgar Riba and K. Mikolajczyk, "Learning local feature descriptors with triplets and shallow convolutional neural networks," in *Proceedings of the British Machine Vision Conference (BMVC)*, E. R. H. Richard C. Wilson and W. A. P. Smith, Eds., BMVA Press, Sep. 2016, pp. 119.1–119.11, ISBN: 1-901725-59-6. DOI: 10.5244/C.30.119. [Online]. Available: https://dx.doi.org/10.5244/C.30.119.

[71] M. Zakariah, "Digital multimedia audio forensics: Past, present and future," *Multimedia Tools and Applications*, vol. 77, Jan. 2018. DOI: 10.1007/s11042-016-4277-2.



NTNU

Kunnskap for en bedre verden