Bachelor thesis

Artificial Intelligence

Radboud University 

Instrument Classification for Cochlear Implant Filtering using Convolutional Neural Networks

Author:
Hidde Jansen
S1006034

Supervisor:
Dr Y. Güçlütürk
Donders Centre for Cognition
y.gucluturk@donders.ru.nl

Second reader:
Dr U. Güçlü
Donders Centre for Cognition
u.guclu@donders.ru.nl

January 26, 2021

Abstract

In this thesis, Convolutional Neural Networks are used to classify instruments in audio files containing musical instruments. In particular, music enjoyment for Cochlear Implant users is central in this thesis. Therefore, there are different frequency filters applied in the preprocessing, to further explore the optimal preprocessing when frequency space is limited. The results show that, in line with previous research, there is a certain range that is particularly important in instrument recognition. However, in this research this range differs, possibly because this research is focused on music whereas the original research is based on speech. The classification when filtered on this range outperforms the classification when presented with the full spectrum.

Contents

Chapter 1

Introduction

Humans use their senses to navigate around in the world and make sense of it. Hearing is one of these senses and it is a very substantial part of our lives. Our hearing has multiple functions, for example it is used while navigating in the form of listening for dangers (cars, other people, etc.). Another example is the use of hearing while relaxing, for example by listening to music. Not all humans can hear in the same way, some humans have trouble hearing and therefore may want to use a hearing aid. One kind of these hearing aids is a Cochlear Implant or CI for short.

1.1 The working of normal human hearing

To understand how a cochlear implant can help in restoring the ability to hear, it is useful to know how normal human hearing works.

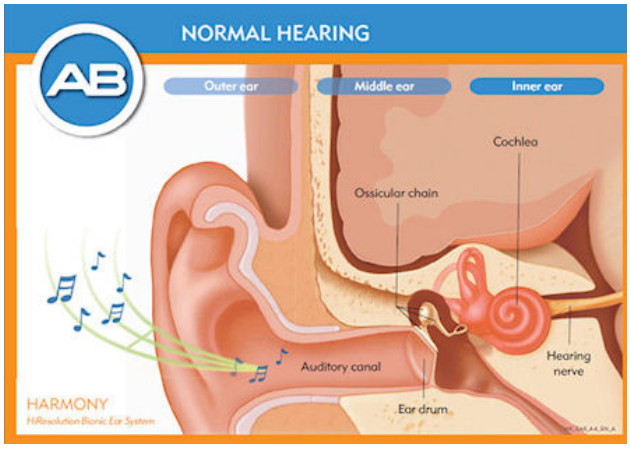


Figure 1.1: Overview of normal hearing[1]

In figure 1.1 a simple view of the inner ear is presented. Audio waves enter the auditory canal, pass through the eardrum, get amplified by the

ossicular chain and ultimately end up in the cochlea. The cochlea is filled with fluid and a large number of hair cells. When sound enters the cochlea it causes the fluid and ultimately the hair cells to vibrate. The vibration of these hair cells then results in a small electrical current, which is passed on through the auditory nerve to the brain. This entire chain should function in order to have a sense of hearing. If a patient has inner-ear damage and the patient can not be helped by an external hearing aid anymore, a cochlear implant may be an option. The cochlear implant bypasses almost the entire chain, to deliver sound signals directly to the auditory nerve. The cochlear implant makes use of a property of the cochlea called cochlear tonotopy. This property means that the cochlea is ordered according to frequency. The start of the cochlea (also called the base) has hair cells that are sensitive to high frequencies while the end of the cochlea (also called the apex) has hair cells that are sensitive to low frequencies.

1.2 The cochlear implant

A typical multi-channel CI consists of a microphone, a preprocessor, a transmitter/receiver and an array of electrodes. This setup can take over almost the whole chain of hearing. The microphone, preprocessor and transmitter are placed externally on or nearby the ear of the patient[2]. These three components take over all functions from the hearing chain before the cochlea. The sound is captured by the microphone, amplified and processed by the preprocessor and finally transmitted by the transmitter. The electrode array is surgically inserted into the cochlea and is connected to a receiver, which is also inserted surgically. The electrode can be seen as a long string, spanning a part of the length of the cochlea. Because the cochlear implant is making use of the cochlear tonotopy, the electrode can precisely stimulate different regions, sensitive to different frequencies. However, the CI can not replace the entire human hearing. While the cochlea is generally able to capture a large spectrum of frequencies, from 20 Hz to 20 kHz [3], CIs are limited to between 12 and 24 electrodes [4], each spanning a small section of the cochlea. Furthermore, it cannot be said that every position of stimulation in the cochlea directly corresponds to a perceived frequency [5]. This makes it harder to apply theories of human hearing directly to hearing with a CI.

1.3 Speech versus music

While the above limitation of a cochlear implant sounds very serious, it has not proven to be a great obstacle for the use of cochlear implants within the field of speech recognition. Within speech recognition, any number above 8 electrodes has not been proven to be beneficial [4]. This does not mean that the actual performance is good at this point. It may also be the case

that, while performance is bad, it does not increase by adding electrodes. However, speech communication is generally satisfactory with the current cochlear implants.[6].

Nevertheless, While speech is important, many CI users show a desire to enjoy music [6]. This is not very successful in CI hearing yet, because of the great variability and complexity that comes with music.

1.4 The presented research

This research will focus on instrument classification. With a good idea of the instruments that can be heard within a piece of music, this may increase the understanding of music by a cochlear implant user. As mentioned before, the number of electrodes within a cochlear implant setup is limited, therefore not allowing a broad spectrum of frequencies to be perceived. Thus, this research will focus on determining optimal frequencies and preprocessing strategies in order to improve instrument classification.

In order to perform this research, deep learning will be used. This is a machine learning technique that is able to process large datasets. Using machine learning for this research means that these algorithms can later be used to assess the applicability of new preprocessing techniques without having the need for human participants. Furthermore, because of the applicability to large datasets, something can already be said about the usability in the real world by using machine learning.

The goal of this research is thus to have a deep learning network that can emulate the human hearing reasonably well and to have evaluated different filtering/preprocessing techniques to raw audio in order to improve instrument classification. The research question that this research will address is defined as:

"What is the optimal preprocessing procedure for instrument classification by CI users?"

My hypothesis is that the optimal preprocessing procedure can not be determined entirely as of now, however filtering on different frequencies will be important. I expect to find an improvement on instrument classification when higher frequencies are filtered out.

1.5 The dataset

In order for deep learning to work effectively, one needs a large enough dataset in order to train the network. This research will mainly focus on the IRMAS dataset [7]. The training set of this dataset consists of 6705 .wav files containing annotated instrument recordings. These recordings may have multiple instruments in them but are then annotated with the most

dominant instrument in the recording. The possible instruments within the recordings are cello (cel), clarinet (cla), flute (flu), acoustic guitar (gac), electric guitar (gel), organ (org), piano (pia), saxophone (sax), trumpet (tru), violin (vio), and human singing voice (voi). Furthermore all audio files are sampled at 44.1 kHz and are in 16 bit stereo.

The rest of this thesis is structured in the following way: In chapter 2 some theoretical background about music processing techniques and deep learning will be discussed. Continuing with chapter 3 related work will be discussed, followed by the detailed approach in chapter 4 and corresponding results in chapter 5. Finally, in chapter 6, conclusions will be drawn from the research.

Chapter 2

Preliminaries

In this chapter the theoretical background and tools will be discussed. In particular, knowledge about signal processing techniques will be reviewed first and after that deep learning will be reviewed.
This section does not introduce new research material and can be skipped if the concepts are familiar.

2.1 Signal Processing

2.1.1 Fourier analysis

The frequency of a sound is a great part of the signal itself. The frequency domain of a time-domain signal can be extracted using the Fourier analysis. Since the audio is sampled, the audio in this research is a discrete timed audio signal. The frequency domain of such a signal can be extracted using Discrete Fourier Transform (DFT). This transforms the signal to samples of the Discrete-time Fourier Transform (DTFT). However, this gives one result spanning the whole audio signal, where the time-domain data is lost. Since this research is about music, there needs to be a way to see the frequencies at each sample. This is were Short Time Fourier Transform or STFT comes in. This is a windowed function which slides over the complete audio signal and determines the Fourier Transform at each individual step. When concatenating these also in the time-domain, the frequencies can be determined for the scope of the whole signal.

2.1.2 Mel-Spectrogram

As opposed to the linear scale of frequencies in the Hz range, humans have perceptual hearing, meaning they cannot distinguish different frequencies over the whole range in the same way. [8]. This notion has been converted into a well-known scale called the mel scale by Stevens, Volkmann and Newmann [8]. This scale makes sure that notions such as a frequency “twice

as high" are actually perceived as twice as high while they are not exactly the double amount of Hz. They do this by converting the linear scale to a logarithmic scale with the following formula:

$$\mathrm{a} = 2595 \log_{10}(1 + \frac{b}{700})$$

Where a is the frequency in "mels" and b is the frequency in Hz. From this formula the actual effect can also be observed. For $b = 1000$ and $b = 1500$ the values for a are $a = 999.98...$ and $a = 1290.55...$ respectively. However, if done for $b = 8000$ and $b = 8500$ the values for a are $a = 2840.02...$ and $a = 2902.99$ respectively. From these examples it follows that the perceptual difference between a frequency of 1000 Hz and 1500 Hz is experienced as about 300 mels, whereas the perceptual difference between a frequency of 8000 Hz and 8500 Hz is experienced as about 60 mels. The absolute difference, 500 Hz, is the same in the two situations. Combining the mel spectrum of a signal with the time domain, there can be formed a spectrogram. This spectrogram is the main way of representing audio in this research.

The specifics and an example of how all preprocessing steps are combined can be found in the appendix in section A.1.

2.2 Deep Learning

In this research, deep learning will be used. More specifically, a convolutional neural network will be used. A convolutional neural network is shortly said a deep learning algorithm which can take an image as input and, by assigning different weights and biases to different aspects of the image, is able to generate filters from images and as such differentiate multiple images from each other [9]. The image in our case will be the mel-spectrogram. The weights and biases are learned in a supervised way, i.e. the network is given a lot of data (the training data) and it is told what the sample represents (for example a cello). The network then tries to learn specific features from the spectrogram that indicates that the sample is a cello. The exact network layout will be discussed in chapter 4.

Chapter 3

Related Work

Relevant related work for this research can be divided into two parts: the instrument classification and the application onto cochlear implants.

3.1 Instrument classification

Sing, Bachhav, Joshi and Patil provided an excellent introduction to instrument classification in their paper "Musical Instrument Recognition using CNN and SVM Prabhjyot" [10]. What they describe are two of the most currently used techniques for instrument classification, namely convolutional neural networks (CNN) and support vector machines (SVM). These techniques are readily used.

For example, Gomez et al discuss a similar approach to instrument classification as is in this research in [11]. Their approach is similar to the approach used in this research because of the mel spectrograms and the use of a convolutional neural network. What is different is the layout of the network and the use of transfer learning. In this research, transfer learning is not used. The way the results are calculated in their research is a bit different compared to mine, but they seem to get an F-score of about 60 %.

Similar research has also been done by Lara Haidar-Ahmad in [12].The research is quite similar, although the network layout is a bit different here again. Furthermore, they added random noise to the signal to prevent overfitting. I have tried to do this by splitting the audio signals and generating more samples in this way. Furthermore, they have cut down the number of classes present in the IRMAS dataset. They achieve an overall recall of 64.50 %.

3.2 Application onto Cochlear Implants

Petrov and Gritsjuk[13] have also experimented with different frequency ranges in cochlear implants, however they had the focus on speech and used

human participants. The research presented in this thesis is inspired by their research. However, the research in this thesis uses music and a convolutional neural network. The proposed frequency filters in [13] are also the filters used in this thesis. Their main conclusion was that the frequency range 250-6500 Hz was the most important for speech recognition.

Chapter 4

Research Methods

In order to say something about the research question a few steps need to be defined: Preprocessing, the network and the evaluation of the process.

4.1 Preprocessing

The programming language Python [14] will be used for the preprocessing. For the IRMAS [7] training dataset the following preprocessing steps need to be performed:

1. The raw audio files will be read using librosa [15], they will automatically be resampled to 22050 Hz as described in section A.1.1.
2. To partially mediate the problem of the small dataset, each 3-second audio fragment is split up into three 1-second audio files. This thus provides three times the number of datapoints as compared to the original dataset.
3. Of each audio fragment, the mel spectrogram is computed. Librosa [15] calculates the necessary STFT automatically.
4. The correct label is contained in the filename and is extracted.
5. Both the spectrogram and label are stored in two separate lists (having a "data" list and a "label" list.
6. To also train on controlled multi-instrument recordings, 4000 random samples are combined and added to the dataset. For each of these combined recordings, the above steps are redone. Recordings are ignored if they contain the same instrument multiple times, to smoothen the learning process.
7. The lists containing the data and labels are converted to NumPy [16] arrays and are saved in the compressed npz format. This makes the

process of transferring the data to the actual computing hardware easier.

4.2 The actual network

The implementation of the neural network is built in Keras [17] in combination with Tensorflow [18]. The network has the following structure:

1. The mel spectrogram will enter the network through a convolution layer with 32 filters, a kernel size of 3 and a stride of 1.

2. Then, batch normalization is added, before a LeakyReLU activation layer and MaxPooling (with a pool size of 3).

3. A 0.6 dropout layer is added and the above steps are repeated for 64 and 128 filters.

4. The network now incorporates a flattening layer, before two dense layers of 512 and 256 units. Both layers use the sigmoid function as their activation function.

5. Finally, to get the probabilities for the 11 classes, a dense layer with 11 units is added, also with a sigmoid activation function. The sigmoid function as opposed to softmax is because of the multi-label classification.

It is also shown in the following diagram:

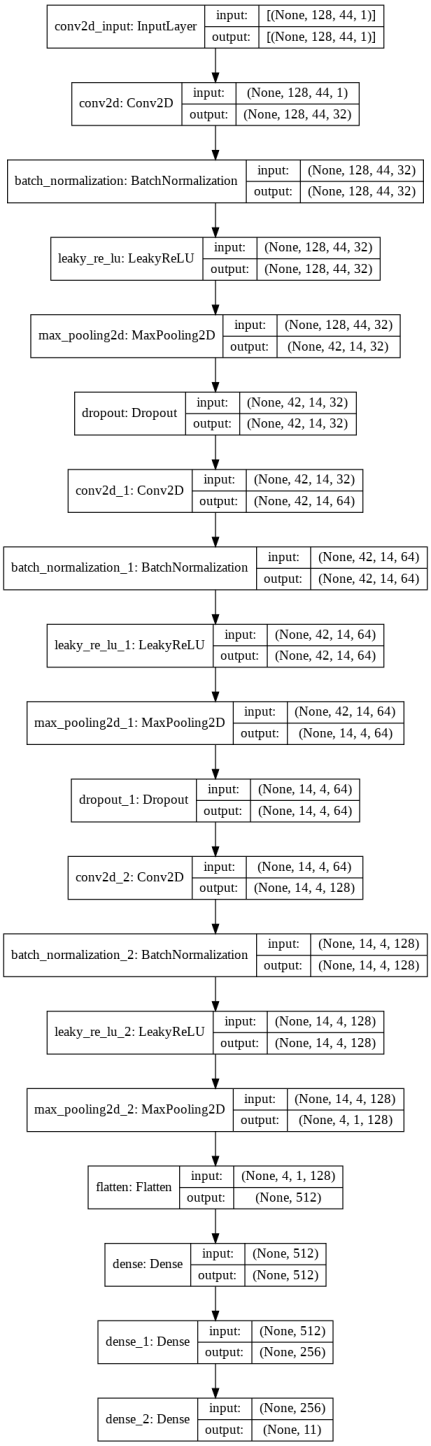


Figure 4.1: Diagram of the network layout

The code for the network can be found in the appendix, section A.2

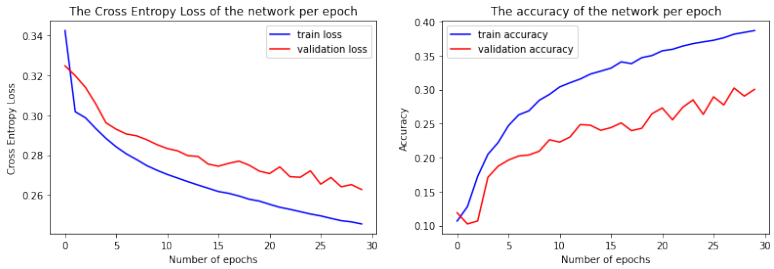
The network is kept quite simple to reduce the risks of overfitting.

4.3 The training

The training set originally contained 6705 single-track recordings, which were split into three 1-second recordings. This gave a total of 20115 recordings. For training, 90 % of the data is used, thus 18103 recordings are used for training. The other 10 % of the data is used for validation. The dataset is not entirely balanced, there are a little less recordings available of cello as compared to the other instruments. However, this is not compensated for since the dataset is already quite small. The training parameters are as follows:

- Loss: Cross-entropy
- Mini-batch size: 128
- Epochs: 30
- Optimizer: Adam
- Learning rate: 10e-3

The loss and accuracy of the training and validation dataset can be seen below.



(a) Cross entropy loss over epochs while training

(b) Accuracy over epochs while training

Figure 4.2: Metrics of the training

It can be seen that the network slightly starts overfitting after the 12th epoch. However, since the loss is still going down the training is stopped at 30 epochs. The accuracy at that point seems reasonable. The training accuracy is about 39 % and the validation accuracy is about 31 %.

4.4 Evaluation and filtering

Now that the network is trained, the network will be evaluated. In order to do this, a fragment of the test-set of IRMAS [7] is used. This test-set contains samples with multiple instruments, annotated with the most dominant instruments.

In order to emulate the limited frequencies available in CI users, the frequency filters as described in paper [13] will be applied. A comb filter will be applied to emulate the research in [13] but also to emulate the limited frequency bands available for CI users. The comb filter is the same in all the frequency ranges, where $f_0 = 50$ Hz and $Q = 30$.

Chapter 5

Results

As mentioned before, the results are determined by means of a subset of the IRMAS [7] test set. This subset contains 1202 samples with multiple instruments within one recording. The same splitting technique as for the training set is also used in the test set. In particular, all test samples are also 1 second in length.

Since it is difficult to determine a sound strategy for determining correctness, this research takes on the following strategy:

- First, every sample is put through the network to get the probability distribution for every class. Since sigmoid activations are used for the final layers, these probabilities do not sum to 1.
- To get a probability for the sample belonging to a certain class, the total probabilities for the sample are summed and then the result is divided by this sum.
- The previous step therefore results in a relative distribution over classes.
- If the value of the previous step is greater than 0.1, the assumption is that the network thinks that the instrument is present.
- Based on the instruments provided by the previous step, an indication of accuracy is calculated. Here there is another assumption, explained in the next step.
- The correctness assumption made states the following two cases:
  - If there is only one instrument present in the sample and the network identified the correct instrument, the sample is considered to be classified correctly.
  - If there are multiple instruments present in the sample, the sample is considered to be classified correctly if the number of correctly classified instruments is greater or equal to 2. This number

is chosen because the number of instruments generally does not exceed 4 and this should give a "higher than chance" indication.

Using the above strategy, the following results were obtained for the subset of the IRMAS [7] test set:

Range (Hz)	**Correct**	**Percentage correct**
Unaltered	634	0.527
350 - 6500	587	0.488
250 - 6500	620	0.516
250 - 8500	698	0.581
70 - 8500	617	0.513

Table 5.1: Classification results on IRMAS testset

The number in the second column represents the number of correctly classified samples according to the above strategy and the percentage in the rightmost column represents the relative correctness on the 1202 samples. E.g. For the unaltered range there have been classified correctly 634 samples, which is $634/1202 = 0.527\%$.

It can be seen that for the different frequency ranges, different results are obtained.

Chapter 6

Conclusions

As can be concluded from table 5.1, when filtered with a comb filter in the frequency range 250-8500 Hz, the classification performance is optimal and even exceeds the unaltered classification performance. This is in line with my hypothesis that filtering will make a difference. What I did not expect to find is that the frequency range from 6500 - 8500 Hz still makes a difference in speech recognition.

The research and corresponding results in this thesis partly support the conclusion made in [13]. Within that research, the authors show that an unfiltered frequency spectrum is not optimal for speech recognition. They found the frequency range 250-6500 Hz to be the most successful in speech recognition. This research found that the range 250 - 8500 Hz is more important for instrument classification. This can be partially explained by the fact that musical instruments may have a different important frequency range than normal human voice. My research supports their conclusion thus by finding that filtered performance exceeds unfiltered performance, but the actual frequency range differs between the two researches.

Furthermore, general instrument classification on the IRMAS [7] dataset was shown in [11] and [12]. Both researches used the same approach as in my research, a convolutional neural network. However, the first also uses transfer learning, which is not used in this research. Both researches use a lower number of classes whereas this research uses all 11. This research supports the mentioned researches by using a convolutional neural network as a valid way of classifying musical instruments in audio samples. The performance of the network presented in this research is lower, due to the fact that compared to [11] a very simple network and no transfer learning was used. Compared to [12], the research in this thesis performed worse, possibly due to the lower number of classes used in their research and a different layout of the network.

Some of the shortcomings of this research include:

- Use of a single dataset. This can be improved by using multiple stan-

dardized datasets.

- Limited testing with different training hyperparameters. This can be experimented with in the future.
- Limited network complexity. This can be experimented with in the future, however the risk of overfitting is still present. To mediate this, techniques such as adding noise should be evaluated.
- Only using one way of representing musical data, namely mel spectrograms. Different techniques may be experimented with in the future.
- Not a sound technique for performance measurement. A more sound technique can be setup in the future, possibly the network's overall performance will increase by incorporating the above steps and therefore it may no longer be necessary to conclude "correct" when more than 2 instruments are present. Instead, it may be an option to conclude correct if all instruments are classified correctly.

What can be concluded from this research is that instrument classification using convolutional neural networks is definitely possible, in line with [11] and [12] although it is not perfect. It might benefit from extra techniques such as transfer learning as proposed in [12] to make it resistant to the changing nature of music. Right now it needs a standardized dataset to work properly, which may not be available in the real world. Furthermore it shows that there is a specific frequency range important in the case of instrument classification. This is in line with my hypothesis, where I proposed that a certain frequency range will be important for instrument classification. However, higher frequencies do make a difference in instrument classification, which is not in line with my hypothesis. With this knowledge, efficient algorithms may be designed to further help preprocessing musical data for patients with a cochlear implant.

Bibliography

[1] "Normal hearing," Oxford University Hospitals NHS Foundation Trust, 2020. [Online]. Available: https://www.ouh.nhs.uk/services/departments/specialist-surgery/audiology/auditory-implants/ear.aspx

[2] "What is a cochlear implant?" Oxford University Hospitals NHS Foundation Trust, 2020. [Online]. Available: https://www.ouh.nhs.uk/services/departments/specialist-surgery/audiology/auditory-implants/cochlear-implant.aspx

[3] L. Robles and M. A. Ruggero, "Mechanics of the mammalian cochlea." *Physiological reviews*, vol. 81, no. 3, pp. 1305–52, jul 2001. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/11427697http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3590856

[4] N. B. H. Croghan, S. I. Duran, and Z. M. Smith, "Re-examining the relationship between number of cochlear implant channels and maximal speech intelligibility," *The Journal of the Acoustical Society of America*, vol. 142, no. 6, pp. EL537–EL543, 2017. [Online]. Available: http://dx.doi.org/10.1121/1.5016044

[5] K. Vermeire, D. M. Landsberger, P. H. Van de Heyning, M. Voormolen, A. Kleine Punte, R. Schatzer, and C. Zierhofer, "Frequency-place map for electrical stimulation in cochlear implants: Change over time." *Hearing research*, vol. 326, no. 1, pp. 8–14, aug 2015. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/25840373http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4524783

[6] E. Kim, H. J. Lee, and H. J. Kim, "Music perception ability of Korean adult cochlear implant listeners," *Clinical and Experimental Otorhinolaryngology*, vol. 5, no. SUPPL. 1, pp. 53–58, 2012.

[7] J. J. Bosch, J. Janer, F. Fuhrmann, and P. Herrera, "A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals," *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012*, no. Ismir, pp. 559–564, 2012.

[8] S. S. Stevens, J. Volkmann, and E. B. Newman, "A Scale for the Measurement of the Psychological Magnitude Pitch," *Journal of the Acoustical Society of America*, vol. 8, no. 3, pp. 185–190, 1937.

[9] S. Saha, "A comprehensive guide to convolutional neural networks - the eli5 way," Dec 2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[10] P. Singh, D. Bachhav, O. Joshi, and N. Patil, "Musical Instrument Recognition using CNN and SVM," *International Research Journal of Engineering and Technology*, pp. 566–569, 2019. [Online]. Available: www.irjet.net

[11] J. S. Gómez, J. Abeßer, and E. Cano, "Jazz solo instrument classification with convolutional neural networks, source separation, and transfer learning," *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018*, pp. 577–584, 2018.

[12] L. Haidar-ahmad, "Music and Instrument Classification using Deep Learning Technics," *CS230*, 2018.

[13] P. SM, "Modeling of Cochlear Implants with Different Frequency Ranges by Means of Spectrally Deprived Speech," *Journal of Otolaryngology-ENT Research*, vol. 6, no. 4, pp. 4–6, apr 2017. [Online]. Available: https://medcraveonline.com/JOENTR/modeling-of-cochlear-implants-with-different-frequency-ranges-by-means-of-spectrally-deprived-
html

[14] G. Van Rossum and F. L. Drake Jr, *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[15] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8, 2015.

[16] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del R'ıo, M. Wiebe, P. Peterson, P. G'erard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2

[17] F. Chollet *et al.*, "Keras," 2015. [Online]. Available: https://github.com/fchollet/keras

[18] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[19] B. Mcfee, Jul 2019. [Online]. Available: https://librosa.org/blog/2019/07/17/resample-on-load/

Appendix A

Appendix

A.1 Example preprocessing

A.1.1 Combining the preprocessing

Since the above theories are quite abstract, here will be shown an example of the preprocessing that is done to every file by means of one example file. This file is part of the IRMAS trainingset and is a three-second long recording of a piano sound with two key actions. Since Librosa is used, there are some specific artifacts which will also be highlighted. The processing starts by loading in the audio file into a numpy array. This is done using Librosa. With the load function of Librosa, the audio will be resampled to 22050 Hz and the signal will also be converted to mono. The first reduces memory consumption while still preserving enough detail and because of the mathematical theorem by Nyquist and Shannon, which says that if above a certain frequency there is no response anymore, the sampling rate can be reduced [19]. The result is a numpy array containing time-series data of the audio signal. In particular, the array contains the amplitude of the signal at each sample.

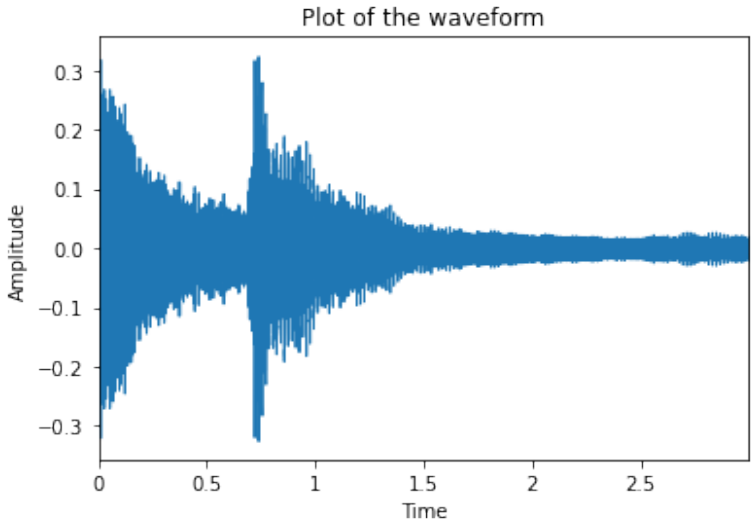
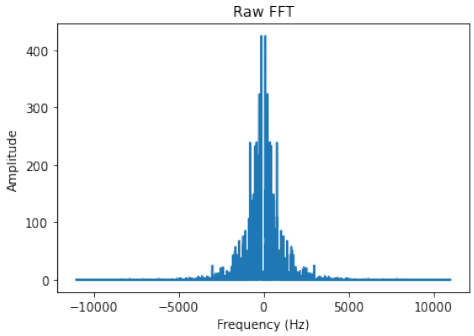


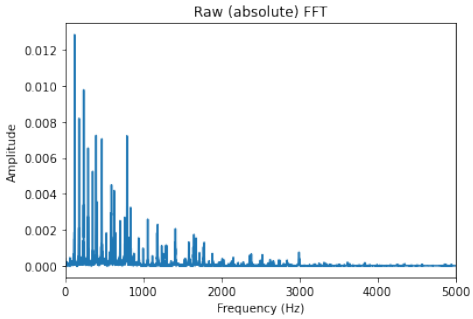
Figure A.1: The waveform of the example audio file

The x-axis has to be converted to seconds instead of samples to make it more understandable. It is quite clear from the plot that the two key actions happen right at the start and at about 0.75 seconds. After that, the audio signal decreases until the 3 seconds are met.

While this waveform gives us some idea of the audio signal, it does not tell us anything about the frequencies in the signal. To convert this time-series data to frequency data, the Fourier transform is used. When directly applying the Discrete Fourier Transform on the data, the result is contained in the following plot.



(a) The raw FFT (DFT) of the example audio file



(b) The cleaned FFT of the example audio file

Figure A.2: FFT of the example audio file

Because of the mathematical properties of the DFT, there also is a negative counterpart. This effect is also present on the y-axis, but that effect is removed for comprehensibility. To further improve the comprehensibility, the frequency range has been trimmed in A.2b. It can be seen that the most activity is between 0 and 2000 Hz. However, DFT can only say something about the signal as a whole, not at individual time points. In order to also incorporate the time-domain into the analysis, STFT is used. The resulting plot is a bit different compared to the plots above:

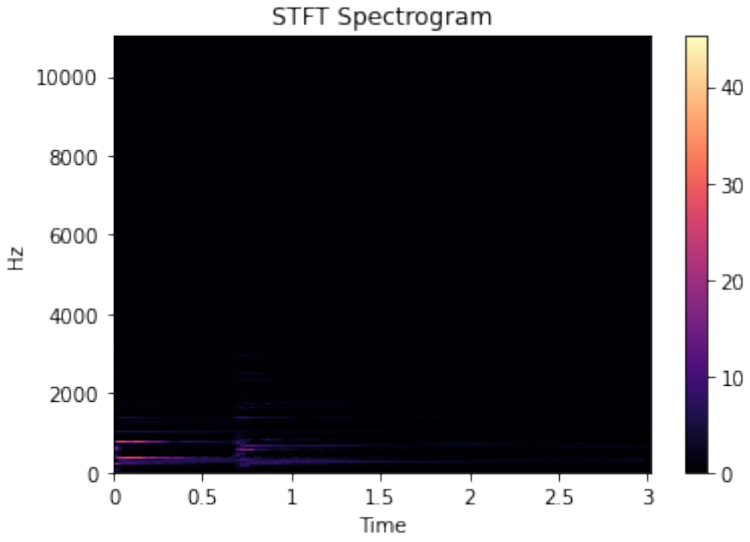


Figure A.3: STFT spectrogram of the example audio file

In the above figure the time-domain is expressed on the x-axis and the frequency is expressed on the y-axis. Furthermore, the color indicates the presence and intensity of a certain frequency at a certain timepoint. Here the presence of the frequencies as indicated in figure A.2 (frequency domain), combined with the waveform as indicated in figure A.1 (time-domain) can be observed. This so-called spectrogram is thus a way to capture both the time-domain and frequency-domain within one figure.

When converting the STFT to the mel scale, the following plot follows:

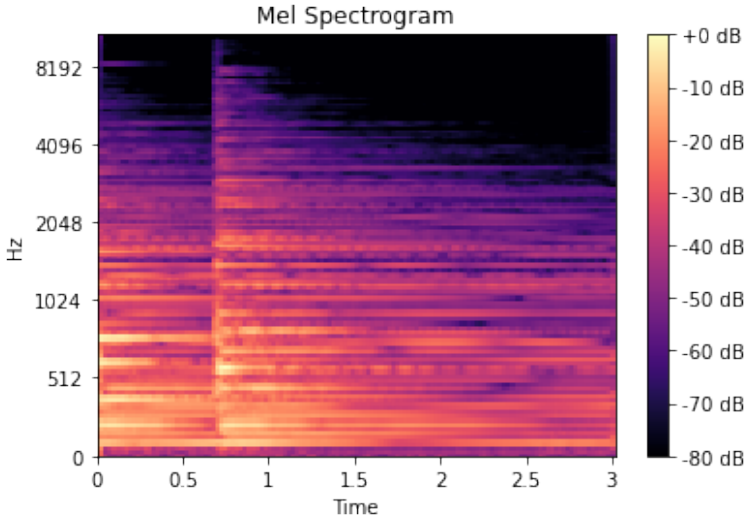


Figure A.4: Mel spectogram of the example audio file

Finally, this is a way to capture all the sound data in a format that is understandable and is perceived correctly according to normal hearing human participants. This format of representing the audio is the basis of the research in this thesis and will be used to train and evaluate the network later on in this research.

A.2 Network

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(3,3), strides=(1, 1), padding='same',
input_shape=(128,44,1), use_bias=False))
model.add(BatchNormalization())
model.add(LeakyReLU())
model.add(MaxPooling2D(pool_size=(3,3)))

model.add(Dropout(0.6))

model.add(Conv2D(64, kernel_size=(3,3), strides=(1, 1), padding='same',
use_bias=False))
model.add(BatchNormalization())
model.add(LeakyReLU())
model.add(MaxPooling2D(pool_size=(3,3)))

model.add(Dropout(0.6))

model.add(Conv2D(128, kernel_size=(3,3), strides=(1, 1), padding='same',
use_bias=False))
model.add(BatchNormalization())
model.add(LeakyReLU())
model.add(MaxPooling2D(pool_size=(3,3)))

model.add(Flatten())

model.add(Dense(512, activation='sigmoid', use_bias=False))
model.add(Dense(256, activation='sigmoid', use_bias=False))

model.add(Dense(11, activation='sigmoid', use_bias=False))
```