What bird is that?

An attempt in using the Wigner-Ville distribution to identify bird songs

Axel Wetterlundh

Supervisor: Maria Sandsten

June 2020

LUND
UNIVERSITY

Faculty of Science

Acknowledgement

I would like to thank my supervisor Maria Sandsten for guiding me through this thesis. I very much appreciate all the good advice and ideas that you gave throughout the process.

Abstract

The presence of birds in an ecosystem is often a good indicator of the overall biodiversity. Since birds can be hard to see, their sounds are often used instead to measure their presence. To automatically detect birds the most common method is to use a time-frequency representation together with a convolutional neural network. The most used time-frequency representation is called the spectrogram. An alternative to this is the Wigner-Ville distribution (WVD). The purpose of this thesis is to investigate if bird classification can be improved if the WVD is used instead of the spectrogram.

The bird sounds were gathered from the website xeno-canto.org. Nine bird species were selected and there were in total 859 samples of bird songs.

To achieve the purpose, four different methods were used. The first one compared the spectrogram to the WVD. The second one compared the spectrogram to the smoothed pseudo Wigner-Ville distribution (SPWVD). The third one compared the spectrogram to several SPWVD's, performed on shorter sound segments. The last one investigated if a high pass filter could improve the methods.

The WVD and its variations performed worse than the spectrogram for all methods. The best result for the spectrogram was 79% while the best result for the WVD came from a variant of the SPWVD. Its maximum accuracy was 70%. The poor performance of the WVD is likely, in part, a result of the high computational requirements for the WVD. As a result of this, much shorter sound segments could be utilised for the WVD compared to the spectrogram. In the future it is likely that the computer power available will far exceed the current availability thus giving the WVD a better chance.

Popular science description

When researchers are assessing the biodiversity of an area, they often look at the birds. This is because a healthy bird population often means a healthy animal and plant life in general. Since birds can be quite hard to see, researchers often use their song to identify the species and how many of them that are present in an area. This process can be quite time consuming and there is a need for automatic methods.

To automate processes like this, one often uses artificial intelligence and machine learning. One of the more commonly used methods for machine learning is called neural networks. For neural networks to work properly they need good data. When it comes to identifying birds using their sound, the neural networks tend to perform better if the data contains the pitch of their sounds, i.e. their frequencies, rather than the sounds themself.

To find the frequencies one often uses time-frequency representations. These representations visualises how the frequencies of a sound signal vary with time. There are many kinds of time-frequency representations but the most used version for bird song identification is called the spectrogram. There are however other time-frequency representations, one popular choice is called the Wigner-Ville distribution (WVD). It gives a clearer visualization than the spectrogram but takes longer for the computer to compute which means that shorter sound segments must be used. The purpose of this thesis is to investigate if the WVD can improve the accuracy when identifying birds compared to the spectrogram.

To achieve this result four different methods of the WVD and the spectrogram were analysed. The results of the spectrogram and the WVD were then compared. The best result for the spectrogram was 79% while the best result for the WVD was 70%. This means that the WVD could not improve the accuracy. One of the shortcomings of the WVD was the computational burden which meant that shorter sound signals had to be used. In the future, when computer power has increased, it is possible that the WVD could achieve better accuracy than the spectrogram.

Contents

1 Introduction **2**
1.1 Background and previous research . . . 2
1.2 Problem formulation and Purpose . . . 3

2 Theory **4**
2.1 Time-frequency representations . . . 4
2.1.1 Spectrogram . . . 4
2.1.2 Wigner-Ville Distribution . . . 5
2.1.3 Smoothed Pseudo Wigner-Ville distribution . . . 6
2.2 Classification . . . 8
2.2.1 Neural Networks . . . 8
2.2.2 Convolutional Neural Networks . . . 9

3 Method and material **10**
3.1 Data . . . 10
3.2 Method . . . 11
3.2.1 Data handling . . . 11
3.2.2 First analysis . . . 11
3.2.3 Second Analysis . . . 11
3.2.4 Third Analysis . . . 11
3.2.5 Fourth Analysis . . . 13
3.3 Software . . . 13
3.3.1 Time-frequency representations . . . 13
3.3.2 Convolutional Neural Network . . . 13
3.4 Hardware . . . 13

4 Results **14**
4.1 Results from the first analysis . . . 14
4.2 Results from the second analysis . . . 15
4.3 Results from the third analysis . . . 16
4.4 Results from the fourth analysis . . . 17

5 Discussion and conclusion **19**
5.1 First Analysis . . . 19
5.2 Second Analysis . . . 19
5.3 Third Analysis . . . 19
5.4 Fourth Analysis . . . 20
5.5 Conclusion . . . 20
5.6 Final discussion and future research . . . 20

6 References **22**

7 Appendix **23**
7.1 Appendix 1: The settings for the first analysis . . . 23
7.2 Appendix 2: The settings for the second analysis . . . 24
7.3 Appendix 3: The settings for the third analysis . . . 25
7.4 Appendix 4: The settings for the fourth analysis . . . 26

1 Introduction

1.1 Background and previous research

The presence of birds in an ecosystem can be a good indicator of the overall biodiversity in that ecosystem [1]. An effective way to find and identify bird species can therefore be very useful in the study of ecosystems.

One commonly used method for bird identification is to use their songs. This is often done manually which is very time consuming and prone to human errors [2]. An alternative to this, which has seen an increase in usage, is to use solutions that can automatically detect and classify bird songs. This method is increasingly being used [3].

The automatic analysis and classification of sound signals is a common problem in machine learning. Some examples of this are products like Apple Siri or Amazon Alexa that uses the users voice to execute commands. When it comes to classifying bird sounds, a lot of research has been done on the topic.

A lot of different methods exist for the classification of sound data and bird sounds. One of the more popular methods is to use a neural network but to feed time-frequency representations of the sounds, rather than the sounds themself into the network. The idea is that different bird species have different frequencies in their song which can be detected with time-frequency representations. The most used time-frequency representation is called the spectrogram. Below are some, among many, of the previous studies that have used this method.

In 2018 Incze et al. [4] tried to classify bird songs by transforming the sound signals to a spectrogram and feeding it into a convolutional neural network (CNN). The sound sequence was first divided into shorter of three second segments with a sampling frequency of 44100 Hz, on which the spectrogram was calculated. The spectrogram had a window length of 448 with a 50% overlap. The CNN was created with Tensor Flow, which is a popular machine learning package. They managed to get 40% accuracy when classifying 10 different species. The birds were obtained from xeno-canto.org.

In 2016 Tóth et al. [5] used a similar approach with CNN's and spectrograms. They used a hamming window with 50% overlap. The main difference between this method and the previous one was that they did not divide the sound signal into shorter segments. Instead they tried to filter out the interesting parts of the spectrogram and perform the analysis on those parts. This means that all sound signals that the spectrogram was calculated on were of different lengths. The network structure was inspired by a previous net called AlexNet.They were able to achieve 40% accuracy when classifying 999 different species. The birds were also obtained from xeno-xanto.org.

In 2018 Sankupellay et al. [6] managed to get 72% accuracy when classifying 46 species with spectrograms and CNN's. The window length of the spectrogram was 256 with 87.5% overlap. The sampling rate was 22 050 Hz. The lengths of the sound segments used were not disclosed. They used a network called ResNet50 which used pre trained weights. The bird songs were gathered from the same source as before, i.e. xeno-canto.org

The accuracy between the previous studies are quite different. However, there are some problems that make comparison difficult. The biggest of these is that not all of the studies disclose what birds species that were used in the experiments. This is a problem since some bird species are likely to be harder to identify than others.

1.2 Problem formulation and Purpose

While there are many different time-frequency representations one can use when trying to classify bird songs, most studies used the spectrogram. I have found very little motivation to why the spectrogram was chosen as the applied time-frequency representation and a reasonable explanation could be that it is simply the easiest one to understand and use. However, another natural choice would be the Wigner-Ville distribution, described more detailed in section 2.1.2.

The purpose of this thesis is to investigate if the use of the Wigner-Ville distribution, and its variants, can improve the classification of bird songs compared to the spectrogram.

2 Theory

2.1 Time-frequency representations

Classical spectral analysis often assumes stationary signals. Stationary means, in part, that the frequencies of the signal does not vary with time. This is in many situations an unreasonable assumption. A bird song can be very complex, it is not hard to see that it typically violates this assumption.

When analysing time-varying signals one often uses time-frequency representations. These methods gives a 3-dimensional representation of the frequency-variation with time of a signal.

2.1.1 Spectrogram

The spectrogram, denoted $S_x(t,f)$, is perhaps the most intuitive time-frequency representation. Much like the periodogram it is based on the Fourier transform. However, instead of calculating the Fourier transform on the whole signal, the signal is windowed, and the Fourier transform is calculated on each window separately. This is called the short-time Fourier transform (STFT) and is denoted $X(t,f)$. The spectrogram, $S_x(t,f)$ is then simply the squared absolute value of $X(t,f)$. The use of several overlapping windows of a length shorter than the signal then allows for different spectral estimates centred around different time points. The result is a separate spectral estimate for each window and hence, the frequencies can vary over time. A short window decreases the frequency resolution while a longer decreases the time resolution. If the resolution decreases then the uncertainty to the exact position of the frequency increases. Formally the STFT and the spectrogram are defined, for the signal $x(t)$, as:

$$X(t,f) = \int_{-\infty}^{\infty} x(t_1)h^*(t_1 - t)e^{-i2\pi f t_1} dt_1$$

where $h(t)$ is a window centred at time t. The spectrogram can therefore be defined as:

$$S_x(t,f) = |X(t,f)|^2.$$

In practice, the signal $x(t)$ usually consists of discrete time-steps $x_1, x_2, ..., x_N$. The measurements are often measured with some sampling distance T such that $x_k = X(kT)$. For the sampling distance T the sampling frequency f_s is equal to $f_s = \frac{1}{T}$. The discrete spectrogram is calculated as:

$$S_x(n,l) = \Big| \sum_{n_1=0}^{N-1} x_{n_1} h^*(n_1 - n + \frac{M}{2}) e^{-i2\pi n_1 \frac{l}{L}} \Big|^2$$

where M denotes the length of window and L denotes the number of frequency values. The spectrogram is relatively computationally cheap compared to other time-frequency algorithms [7].

Plotted below is a spectrogram of two sine waves with frequency 1 kHz and 0.9 kHz respectively. The length of the sine waves are 2 000 data points. The window length is set to 256 with 50% overlap.

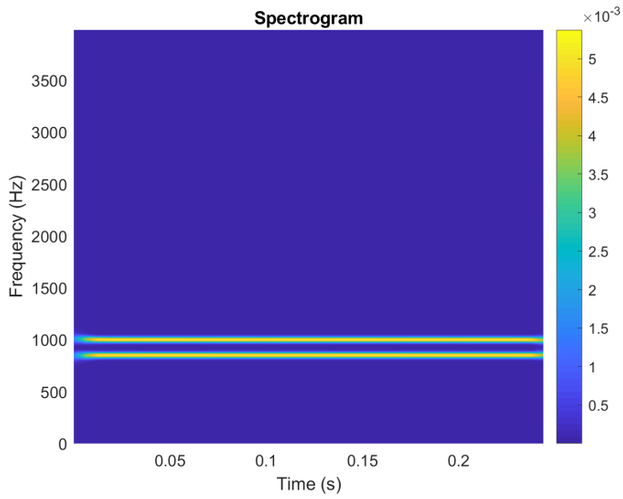


Figure 1: Spectrogram of two sine-waves with frequencies 1 and 0.9 kHz

To visualise what the spectrogram looks like for time varying signals, the spectrogram of a convex quadratic chirp is plotted below. The length of it is 2 000 data points. The window length is set to 32 with 80% overlap. The plot clearly shows how the frequencies decrease from 400 Hz down towards 0 Hz.

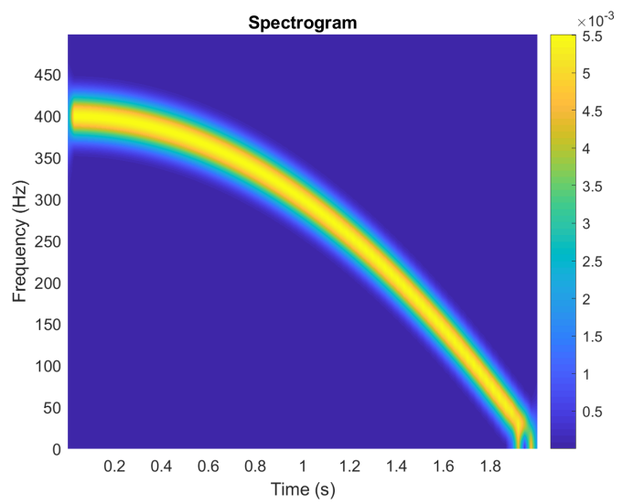


Figure 2: Spectrogram of a convex quadratic chirp

2.1.2 Wigner-Ville Distribution

As mentioned earlier, the spectrogram is not the only time-frequency representation. Another popular choice is the Wigner-Ville distribution (WVD). The WVD for a deterministic signal is defined as:

$$W_X(t,f) = \int_{-\infty}^{\infty} x(t+\frac{\tau}{2})x^*(t-\frac{\tau}{2})e^{-i2\pi f\tau}d\tau.$$

As with the spectrogram, the data sequence $x(t)$ is often divided into discrete time steps $x_1, x_2, ..., x_N$. Therefore, the discrete WVD is often used. It is defined as:

$$W_x[n,l] = \sum_{m=-min(n,N-1-1)}^{min(n,N-1-n)} x_{n+m}x^*_{n-m}e^{-i2\pi m\frac{l}{L}}$$

where L denotes the number of frequency points.

If the signal is real valued, it is often transformed into an analytic signal with the Hilbert transform. This is because the spectrum of the analytic signal is zero for negative frequencies.
The WVD has the advantage of the highest frequency concentration. It is, however, much more computationally expensive than the spectrogram since the auto-correlation function needs to be calculated for each pair of data points. On the hardware setup, used for this thesis, the maximum length of a signal is around 1 000 data points before the memory runs out. Another downside is the presence of cross-terms in between signals [8].

Plotted below is a WVD of the same sine waves as before but shortened to 1 000 data points. The cross-terms between the signals are clearly visible but so is the high concentration of frequency.

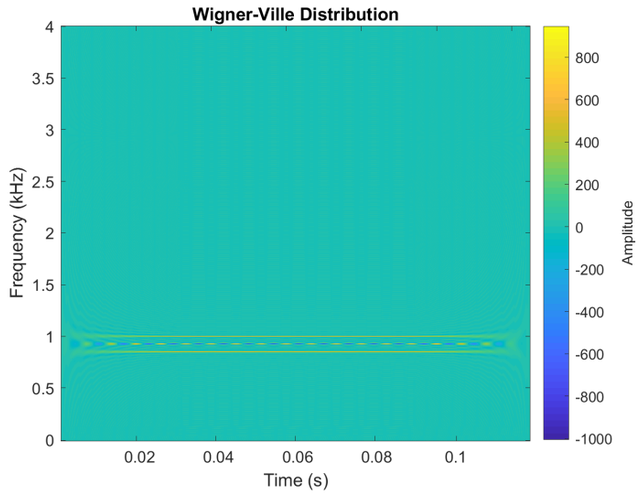


Figure 3: WVD of two sine-waves with frequencies 1 and 0.9 kHz

As with the spectrogram, to visualise what the WVD look like for a time varying signal, the WVD of a convex quadratic chirp is plotted below. The chirp is 1000 data points, i.e. half of the length of the one used for the spectrogram. The high frequency concentration is clearly visible.

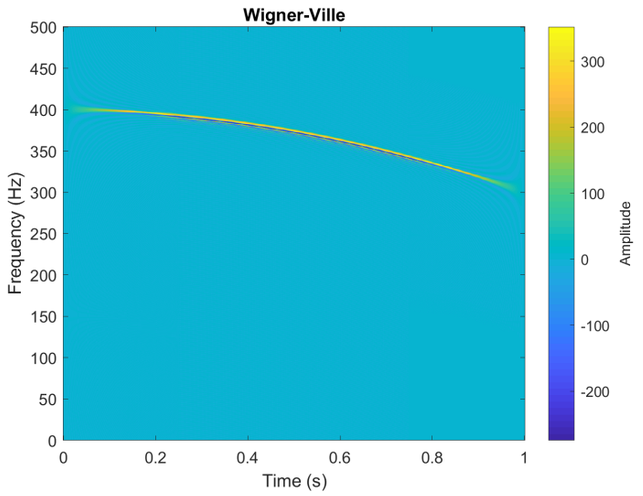


Figure 4: WVD of a convex quadratic chirp

2.1.3 Smoothed Pseudo Wigner-Ville distribution

To deal with the computational limits of the WVD, one can instead use the smoothed pseudo Wigner-Ville distribution (SPWVD) which is defined as:

$$\tilde{W}(t,f) = \int_{-\infty}^{\infty} q(t-u) \int_{-\infty}^{\infty} h(\tau) x(t+\frac{\tau}{2}) x(t-\frac{\tau}{2}) e^{-j2\pi\tau f} d\tau du$$

where h and q are smoothing window functions [9].

The computation time can thus be reduced with a window length shorter than the data sequence. This has the downside of lower frequency concentration. The number of $time-$ and $frequency\ points$ (denoted fp and tp) determine the resolution of the representation, which affect the ability to detect signal components close in time or frequency.

Plotted below are two SPWVD of the a signal containing two sine waves with frequency 1000 and 800 Hz. The dimensions of the representations are $fp \times tp$ which is the only thing that differ between the representations. Distribution 1 (figure A) has higher number of frequency- and time points than distribution 2 (Figure B). The decrease in resolution makes it impossible to see that the signal has two components.

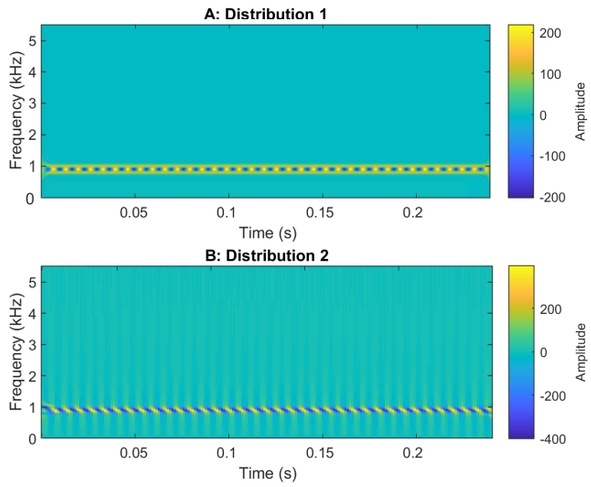


Figure 5: Two plots of two sine waves with frequencies 1 and 0.9 kHz. Figure A has 1999 fp 2000 tp and Figure B has 401 fp 402 tp

Below are four SPWVD of the same convex quadratic chirp as before, with 2 000 data points. The tp and fp are the same as before. A similar decrease in resolution is visible here as well.

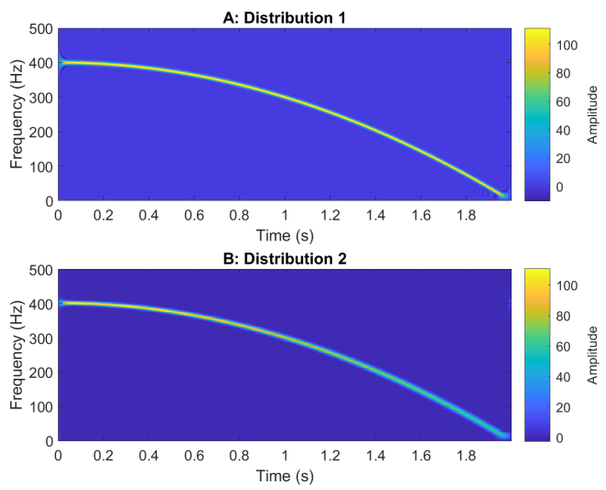


Figure 6: Two plots of a convex quadratic chirp. Figure 5.A has 1999 fp 2000 tp and Figure 5.B has201 fp 2000 tp

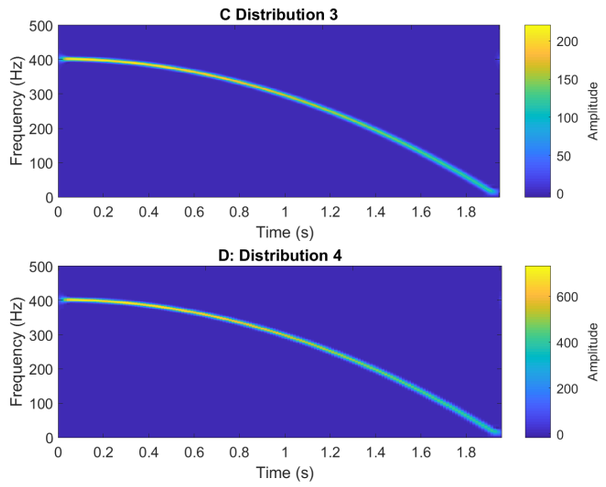


Figure 7: Two plots of a convex quadratic chirp. Figure has 201 fp 1000 tp and Figure D has 201 fp 202 tp

2.2 Classification

Since the sound signals have been transformed into time-frequency representations, i.e. images, there is a need for an algorithm that can classify images. When faced with a classification problem one often uses a machine learning algorithm as a solution. There are many kinds of machine learning algorithm but one of the more common one for classifying images are convolutional neural networks (CNN) which is a subset of the more general neural networks [10]. Below is a description of a general neural network followed by a description of CNN's.

2.2.1 Neural Networks

The goal for a neural network is to, with the function f, map the input x to a category y so that $f(x) = y$. However, the function f is usually unknown so with the parameters θ the wish is to approximate f with $f^*(x;\theta) = y^*$, where x is the input data and y^* is the estimated output data.

The function $f^*(x;\theta)$ is usually structured as a chain such that:

$$f^*(x;\theta) = f^n(f^{n-1}(...f^2(f^1(x;\theta))))$$

where each $f^k(x;\theta)$, for $k = 1...n$, is called a layer. The layer f^1 is called the input layer and f^n is called the output layer. Each layer consists of a vector of size n_k where each number in the vector is called a node. Node j in layer k is denoted $\psi_{j,k}$. The input layer typically only consists of the input data. All nodes in layer k are connected to all nodes in layer $k+1$. Attached to each layer is a subset of the parameters θ denoted θ_k. The parameter θ_k is further divided into the subset θ_k^j for each node j in layer k. The layers between the input and the output layer are called hidden layers. In practice, it works so that $\psi_{j,k+1}$ is the normalised weighted sum of the nodes of layer k, i.e:

$$\psi_{j,k+1} = \sigma\Big(\sum_{i=1}^{n_k} \psi_{j,k}\theta_{i,k}^j\Big)$$

where σ is a normalizing function.

A neural network is trained by being fed input data x with output data y attached to it. The weights θ are then updated with an iterative process called back propagation to minimise a cost function, typically the squared errors, $\sum_{i=1}^{n}(y_i^* - y_i)^2$ [11].

2.2.2 Convolutional Neural Networks

A convolutional neural network is a network with a structure that make it suitable for classifying images. It takes an $N \times L$ array as an input and it has a few extra layers that a general neural network do not have.

The first layer is called a convolution layer. It takes the input array and chooses k number of $j \times j$ cuts from which it feeds to the next layer. Below is a convolution layer with 3 cuts of size 3×4.

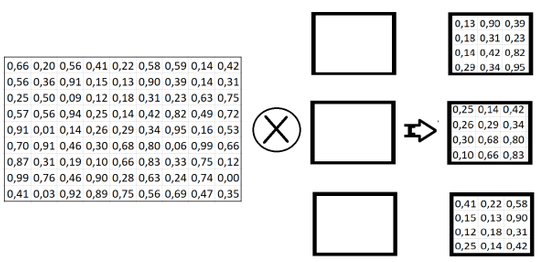


Figure 8: Illustration of the convolution layer

The next layer is called a Relu layer which is a normalising layer. It takes the cuts from the previous layer and normalises it.

The next layer is called a max pooling layer. It windows the normalised cuts from the array and will only give the largest number in each window as an output. This layer has two parameters, stride, and size. Stride is how many data points the window slides each step. The size is the dimensions of the window.

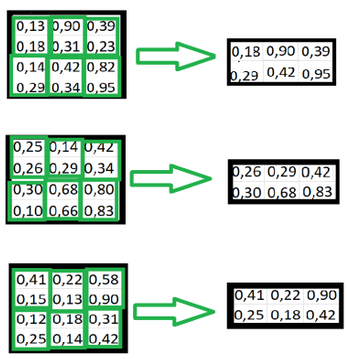


Figure 9: Illustration of the max pooling layer

The resulting arrays are then stacked in a single vector and the rest of the network works as a normal neural network as described above.

3 Method and material

3.1 Data

The data was taken from the website xeno-canto.org [12]. The website holds several hundreds of thousands of sound recordings from more than 10 000 bird species. Due to the size of the database some selection was needed, and 9 bird species were selected. They were selected because of that they sound quite different overall while some species sound very similar pairwise. The website xeno-canto also offer different kinds of bird songs such as calls, songs and alarm calls. Only songs were collected. The training set contained 60% of the data, the validation set contained 20% of the data and the test set contained 20% of the data. Below are two tables. Table 1 show all the birds selected and Table 2 show the birds with pairwise similarity.

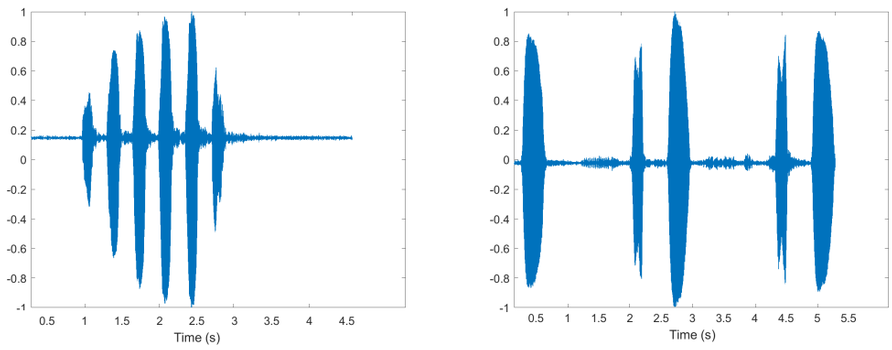
	Number of samples
Boreal owl (*Aegolius funereus*)	100
Cuckoo (*Cacomantis*)	117
Grasshopper Warbler (*Locustella naevia*)	92
Great Reed Warbler (*Acrocephalus arundinaceus*)	95
Nightingale (*Luscinia megarhynchos*)	92
Reed Warbler (*Acrocephalus*)	92
River Warbler (*Locustella fluviatilis*)	93
Tawny owl (*Strix aluco*)	88
Trush Nightingale (*Luscinia luscinia*)	90

Table 1: The birds analysed

Bird	Similar to
Trush Nightingale	Nightingale
Great Reed Warbler	Reed Warbler
Grasshopper Warbler	River Warbler

Table 2: The pairwise similarity between birds

Below are two sound waves plotted where (a) depicts a boreal owl and (b) depicts a cuckoo. They were both sampled at 44 100 Hz and are roughly about 5 seconds each.



(a) The sound of a boreal owl

(b) The sound of a cuckoo

Figure 10: Comparison of two bird songs

3.2 Method

3.2.1 Data handling

The sound files in the chosen data set were of different sampling frequencies, the first step was therefore to re-sample everything to 11 025 Hz. The reason for choosing this frequency was that for most samples it led to an even decrease of a factor four. Furthermore, I did not expect any frequencies larger than 5 500 Hz which is the maximum to avoid aliasing. The second problem was a result from that the sound files were of different lengths, and often too long. To solve this the highest intensity of each sample was found and a symmetric part was cut out around this point. The rest of the analysis was done on this part. All these cut out parts were then manually investigated, and any corrupt samples were thrown away. The data was then transformed to an analytic signal with the Hilbert transform. The spectrogram was calculated on the real valued signal while the WVD was calculated on the analytic signal.

3.2.2 First analysis

Since the purpose of the thesis was to compare the performance of the spectrogram to the WVD, a natural start was to simply compare their performance in a CNN. There were a few parameters for both representations. The first parameter to deal with was the length of the cut out. The length of the cut needed to be optimized separately for both representations. There were however some limits to how long it could be before encountering computational difficulties. The spectrogram could easily handle a data length of 30 000 data points while the WVD had a limit of around 800 data points. The cut lengths started at 30 000 and went down as long as reasonable results were obtained. Other than that, the spectrogram had two extra parameters, window length and overlap. They too needed to be optimised.

Another problem was the dimensions of the representations. The dimensions of the spectrogram could easily be controlled with the window length. The WVD representation was, with the software used, by default of dimensions $N \times 2N$, where N is the length of the sequence. This resulted in a much larger image for the networks to handle. These problems put the WVD at an obvious disadvantage to start with.

Due to these factors the spectrogram was able to handle longer data lengths compared to the WVD. This analysis also compared the performance of the algorithm for different data lengths to see how the performance differed.

Finally, the neural network needed to be optimised. Instead of using the same trained network for both representations, the neural network was instead optimised for each representation separately.

3.2.3 Second Analysis

The second approach was to use the SPWVD representation instead of the regular WVD. This added two more parameters, the number of $frequency\ points$ and the number of $time\ points$ (again denoted fp and tp). The software that was used required the number of fp to be larger than 10% of the data length but smaller than the number of tp. Using this method, a representation could be performed in a reasonable time on a sequence of around 30 000 data points. However, the representation had a dimension of $fp \times tp$. This severely limited how long the data sequence could be. As before the representations were fed into CNN's which again needed to be optimised.

3.2.4 Third Analysis

The third analysis tried to find a way to utilise a longer data sequence. After looking at a lot of time-frequency representations it was clear that the figures more often than not contained a lot of quiet areas. The idea was to begin with the whole sequence but only use the interesting parts of it.

A longer data sequence was therefore divided into shorter segments. These segments will be called the *initial cuts*. The SPWVD was then calculated separately for each *initial cut*. The highest intensity for each representation was then found and symmetric parts were cut out around these points. These parts will be called the *second cuts*. The *second cuts* were then stacked together and used as input in the CNN's. This method added two extra parameters. The first parameter was the lengths of the *initial cut* where shorter cuts lead to more time-frequency representations. The second parameter was the length of the *second cut* where a shorter cut lead to fewer data points that were fed to the network.

Below is an illustrative example where three *initial cuts* are performed on a boreal owl. The *second cuts* are then performed and finally put together. The final figure depicts the end result.

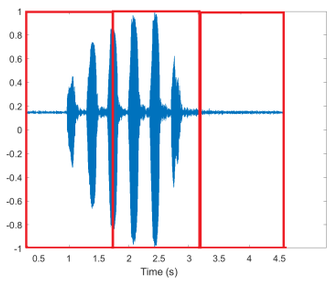


Figure 11: *Three inital cuts on a boreal owl*

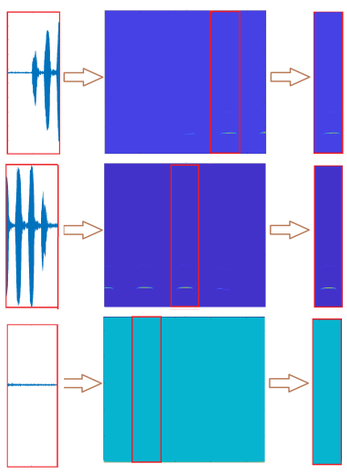


Figure 12: *The highest intensities are found and the second cuts are done around this part. One can see that the third cut contains very little information*

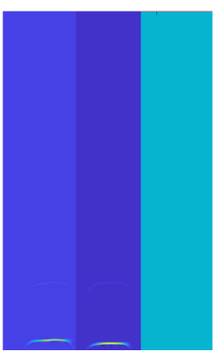


Figure 13: *The final image is formed*

3.2.5 Fourth Analysis

The fourth idea was to use a high pass filter to filter out the lower frequencies, which is likely to be background noise. The idea here was that the WVD might be more prone to noise related errors due to the cross terms which are present between the signal and the noise. This added an extra parameter to be optimised since the results varied depending on what level the high pass filter was set to.

3.3 Software

3.3.1 Time-frequency representations

All time-frequency representations were implemented in MATLAB. The spectrogram was implemented using the function mtspectrogram by Maria Sandsten. This function has the following parameters:

- Window size and type
- Sample frequency
- Number of frequency points
- Number of steps to next frame
- Weights

Only the Hanning window was used which means that the window parameter only determined its length. Furthermore, the weights were only set to their default value, which was 1.

The WVD and the SPWVD was implemented using the MATLAB function wvd. This function has the following parameters:

- Sample frequency
- Number of frequency points (only for SPWVD)
- Number of time points (only for SPWVD)

3.3.2 Convolutional Neural Network

The CNN was implemented with the MATLAB functions trainNetwork and classify. Before these functions could be used the structure of the network and the training options needed to be specified. The following factors needed to be specified:

- Number and size of cuts in convolution layer
- What kind of normalisation should be performed
- Size and number of steps of the max pooling layer
- Depth and size of the hidden layers
- Number of epochs
- Learning rate

Due to hardware limitations, the CNN's could only handle images of around 1 500 000 data points.

3.4 Hardware

All analysis was performed on a HP Elitebook 1030 G3. It had the following specifications.

- Core i5 Processor
- 8 GB RAM

4 Results

4.1 Results from the first analysis

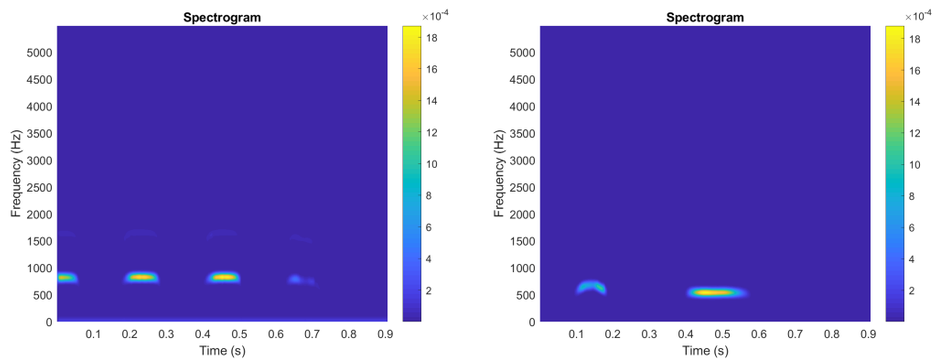
The maximum data length used for the spectrogram was 30 000 data points. With this data length, a maximum accuracy of 75% was obtained. Factors that had an impact on the accuracy was window length, number of frequency points and the structure of the network. The computation time for calculating the spectrograms and training the network was around 30 minutes.

The table below lists the maximum accuracy obtained for different data lengths. For the full description of the set up used for these results, please see Appendix 1.

Data length	Maximum accuracy	Computation time
30 000	75%	30 min
20 000	73%	20 min
10 000	74%	16 min
5 000	70%	15 min
2 500	70%	6 min
1 500	67%	5 min
800	61%	5 min

Table 3: Maximum accuracy of the spectrogram for different data lengths

Below are two spectrograms of two different bird species where (a) is from a boreal owl and (b) is from a cuckoo. The data length was 10 000 which enabled quite complicated frequency variations in time. Both estimates had medium high frequency concentration and are relatively noise free. The noise level for these birds were quite representative for most of the data sets. There were however examples with more noise present.



(a) Spectrogram of a boreal owl

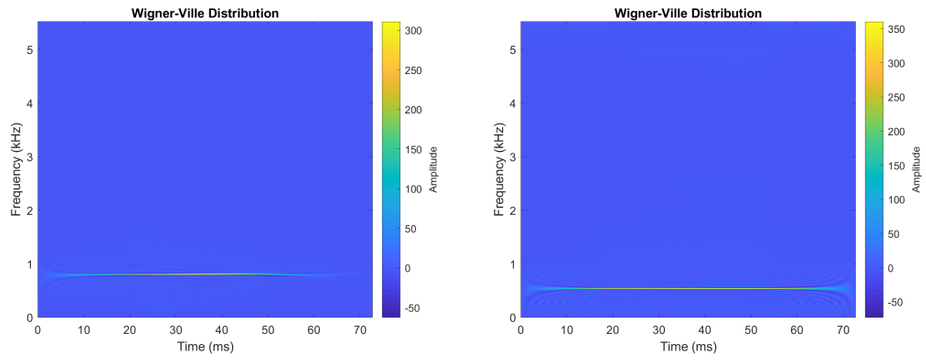
(b) Spectrogram of a cuckoo

Figure 14: Two spectrograms of two birds

The maximum data length used for the WVD was 800 data points, any data length longer than that was not possible due to memory limitations. It had a maximum accuracy of 60% which was around the same as the spectrogram had on the same data length. It took around 5 hours and 30 minutes to calculate the distributions and train the network which was a lot longer than for the spectrogram.

A data length of 600 data points was also used. It managed to get a maximum accuracy of 20%. It took around 3 hours and 20 minutes to calculate the distributions and train the network.

Below are two WVD's of the same birds as above. The 800 data points were used for both distributions. One can see that the spectral concentration was really good but the short data length prevented this method to capture the more complex time variability.



(a) WVD of a boreal owl

(b) WVD of a cuckoo

Figure 15: Two WVD's of two birds

4.2 Results from the second analysis

As described in the method part, two extra parameters were added in this analysis, the number of frequency points and the number of time points. The number of time points needed to be larger than the number of frequency points. The best results were achieved when a long data sequence in combination high amount of time points was used. An example of this was for the data length of 5 000 were the best result came when 1 700 time points were used. A similar relationship held for the number of frequency points where an increase tended to give better results. Another factor that had an impact on the results was the structure of the network.

Due to the hardware limitations the maximum data length that was able to run was 10 000 data points. A longer sequence than that led to too large dimensions which required more memory than was accessible when the networks were trained.

The table below shows the maximum accuracy for the different data lengths. For more detailed information about how the parameters were tuned to achieve these results, please see Appendix 2.

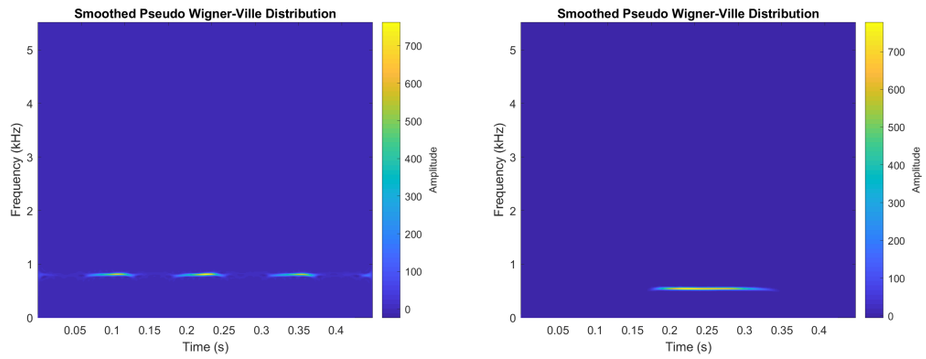
Data length	Maximum accuracy	Computation time
30 000	X	
20 000	X	
10 000	30%	4 hours 40 minutes
5 000	68 %	1 hour 35 minutes
2 500	70 %	1 hour 36 minutes
1 500	69%	5 hour 45 minutes
800	64%	20 min

Table 4: Maximum accuracy and the time it took to calculate the distributions and train the networks for different data lengths

One can see that the best result came when a sequence of 5 000 data points was used. The maximum accuracy did not surpass the maximum accuracy of the spectrogram.

The reason for the extremely long computation time for the data length of 1 500 was that this short data length enabled the use of a larger neural network which in turn took very long to train.

Below are two SPWVD of the same birds as before. The data length was now 5 000 which enabled more complex time variations to be included. One can observe in (a) that it suffered from a bit of cross terms between the components.



(a) SPWVD of a boreal owl (b) SPWVD of a cuckoo

Figure 16: SPWVD two birds

4.3 Results from the third analysis

Since the goal of this analysis was to see if it was possible to utilise a longer data sequence, all analysis was performed on a data sequence containing 30 000 data points. For all initial cuts, the best performance was achieved when the second cut lengths were maximized.

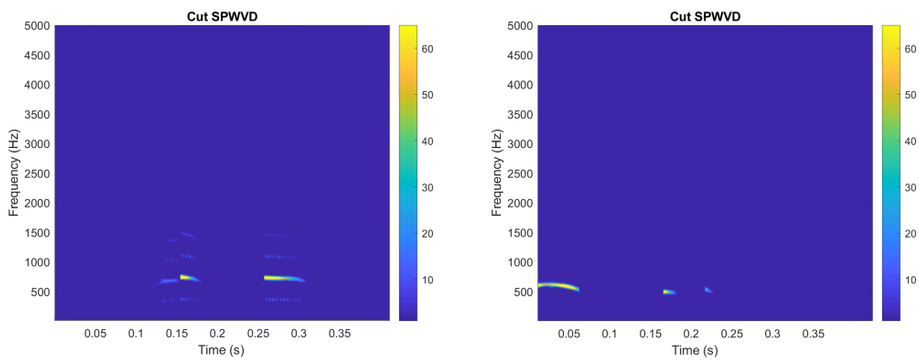
Below are the results of the experiments. One can note that there were no improvements in the results and that the computation times were very long. For the full details on how these results were achieved, please see Appendix 3.

Initial cut length / Second cut length	Maximum Accuracy	Computation time
4 000/350	60%	3 hours 22 minutes
3 500/300	63%	4 hours 35 minutes
2 900/280	61%	5 hours 10 minutes
2 500/240	52%	6 hour 4 minutes

Table 5: Maximum accuracy for the third analysis

Initial cut lengths shorter than 2 500 gave maximum accuracies between 15 and 40% and were therefore not included in the table.

Below are the cut and pasted SPWVD of the same birds as before. The initial cut was set to 3 500 and the second cut to 300.



(a) Cut and pasted SPWVD of a boreal owl (b) Cut and pasted SPWVD

Figure 17: Two cut and pasted SPWVD's of two birds.

4.4 Results from the fourth analysis

The goal of the fourth analysis was to see if a high pass filter could improve the results for the spectrogram or the SPWVD. The table below show the maximum accuracies obtained for the spectrogram and the SPWVD together with the pass bound frequencies that achieved the best results. One can see quite large improvements for the spectrogram but hardly any for the SPWVD. The full description on how these results were achieved can be found in Appendix 4.

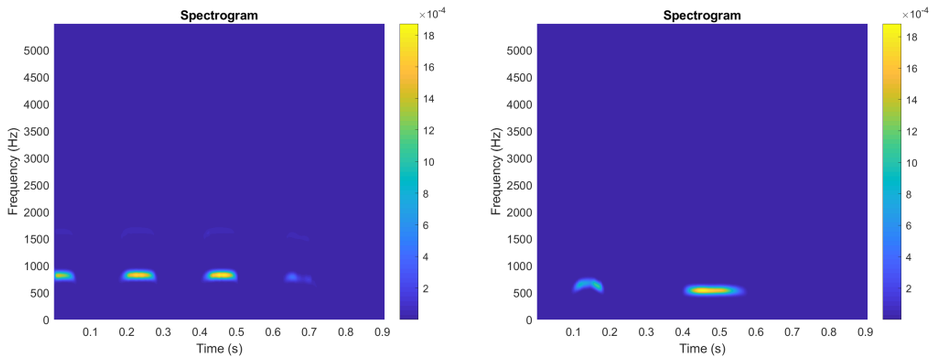
Data length	Maximum accuracy	Pass bound frequency
30 000	79%	500
20 000	77%	500
10 000	74%	500
5 000	79%	500
2 500	73%	500
1 500	67%	500

Table 6: Maximum accuracy of the spectrogram for different data lengths

Data length	Maximum accuracy	Pass bound frequency
10 000	30%	500
5 000	69%	300
2 500	70%	500
1 500	69%	500

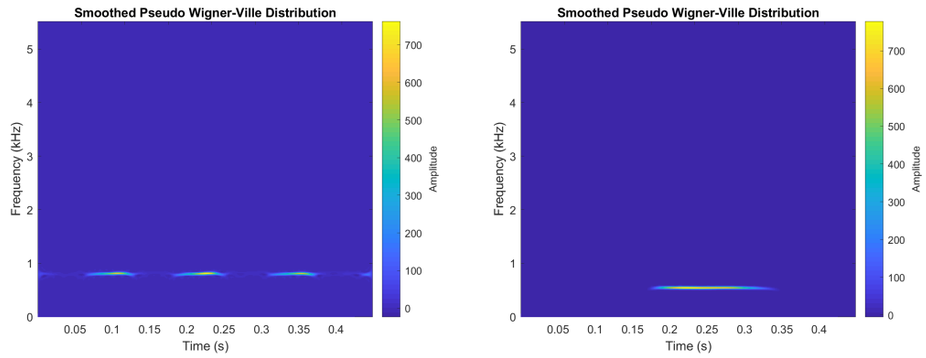
Table 7: Maximum accuracy of the SPWVD for different data lengths

Below are the spectrogram and the SPWVD of the two birds from before. The settings were identical to the first and second analysis except that a high pass filter was added that removed frequencies below 500 Hz. Visually they were very similar to the ones without the filter.



(a) Spectrogram of a boreowl (b) Spectrogram of a cuckoo

Figure 18: Two spectrograms of the two birds with after a high pass filter was applied.



(a) SPWVD of a boreowl (b) SPWVD of a cuckoo

Figure 19: Two SPWVD's of the two birds after a high pass filter was applied.

5 Discussion and conclusion

5.1 First Analysis

Due to the high hardware requirements of the WVD, only one experiment was performed with the same data length as the spectrogram, which was 800 data points. In this scenario both methods performed very similarly with a slight advantage for the spectrogram. However, the reruns with another test set showed quite high variability. This variability was not nearly as high for longer data sequences. It makes the result from this experiment quite difficult to draw any conclusions from. Another important point is that the network used for the WVD was a lot smaller than the one used for the spectrogram. A larger network might have improved the results for the WVD.

The spectrogram worked on longer data sequences and had a higher maximum accuracy. This leads me to believe that it is more important to capture more time variation of the frequencies rather than having a high frequency concentration. Another very appealing aspect of the spectrogram was the computation time. With proper windowing, the dimension of the representation is a lot lower than for the WVD which enables faster training times for the CNN.

An interesting observation is that a longer data length does not necessarily lead to that much better results. When using 10 000 data points the result was only one percentage point lower than the result when 30 000 data points were used. This is especially interesting for the WVD since this means that it might not be that far in the future before we can utilise sufficient data lengths for the WVD.

An obvious factor here is the need for better hardware. To properly draw any conclusion about the viability of the WVD in this setting, better computer power is needed. Analysing the figures of the two WVD's one can see the high frequency concentration. It was a lot more concentrated than the spectrogram. It is therefore a very promising method.

5.2 Second Analysis

The results show that the SPWVD did not surpass the spectrogram in quality. We know from the theory that the resolution increases with a high amount of frequency- and time points. The empirical evidence supports this as the best results were obtained with a high number of frequency- and time points. The number of frequency- and time points was however heavily limited by the hardware which probably led to lower accuracy.

The results showed that the best results came when the data length was 2 500 data points. This was most likely the sweet spot where enough information was present and the resolution of the SPWVD was high enough.

It is likely that a higher number of frequency- and time points would have increased the performance. Given how close the performance was to spectrogram it could very well had surpassed it with a higher number of time and frequency points.

Another factor that might have affected the results in a negative way was the structure of the networks. Due to the high dimensions of the representations, the hardware could not handle large networks. The networks for the spectrogram had deeper layers and more filters in the convolutional layer. If the networks for the SPWVD had similar networks, the results might have been better.

5.3 Third Analysis

The results from this analysis did not surpass the maximum accuracy of the spectrogram. It was also a very time-consuming method with long computational times to calculate the distributions and

train the networks.

The poor performance was likely a result from variability in the data. This method assumed that the bird songs were structured in such a similar way that each initial cut contains roughly the same part of the song for each bird species. This turned out to be an unreasonable assumption and hence the method performed poorly.

Analysing Figure 18, it pretty much confirms this thought. They look quite irregular. A better idea might have been to cut in frequency instead of cutting in time. The figures depict a lot of dead space when it comes to frequencies.

5.4 Fourth Analysis

The results for the spectrogram were very successful as it performed better or at least equally as good for all data lengths. The very best performance increased from 75% to 79%.

The SPWVD improved very little or not at all. This was disappointing since I expected this method to remove the cross terms between the signal and the noise. One explanation for this might be that the poor resolution smeared the frequencies together so that the lower frequencies might already have disappeared.

The images show very little difference between the unfiltered and filtered method so it might be surprising that there were differences at all between the results. There were, however other samples, not shown in any figures, that were nosier and hence the method would have filtered these.

5.5 Conclusion

The purpose of the thesis was to investigate if the use of the WVD, and its variants, could improve the classification of bird songs compared to the spectrogram. While it is clear that I could not achieve this, I do believe that the WVD have great potential. With hardware good enough to utilize a longer data length and larger networks it is likely that the WVD would give the spectrogram a run for its money.

5.6 Final discussion and future research

An interesting observation that I made was just how short the data sequence could be and still yield good results. When I started with the experiments, I assumed that 30 000 data points would be the bare minimum. This turned out to be false and I was able to get good results from much shorter sequences.

As mentioned in the discussion above, the high dimensions of the WVD and the SPWVD forced me to limit how large the networks could be. This in turn raises some more questions. Would the WVD and SPWVD with the current data lengths perform better than the spectrogram if only a larger network had been used? I find this quite unlikely. While changing the hyper parameters in the network had some effect on the results, much bigger effects on the results were achieved when the parameters of the time-frequency representations were changed.

One solution of the problems with the dimensions could be to cut a bit in the representations. There is a possibility that these images contain a lot of dead space which could be disregarded.

The implementation of the WVD and SPWVD had some problems. The functions used were standard MATLAB functions and they offered little ways to customise them. Instead of choosing the number of frequency points and time points a better way would probably be to adjust the window sizes and overlap. A future project would therefore be to implement a function with these

features.

While there are many kinds of classification algorithms, I did only use CNN's. An interesting study would be to use other classification algorithms, such as support vector machines. Support vector machines are not as computationally heavy as CNN's. This could mean that longer data sequences could be utilised which is especially interesting for the WVD. A possible future study would be to compare the spectrogram and CNN's with the WVD and support vector machines.

There are also other kinds of time-frequency representations that could have been used instead of the WVD. One of these methods is called the ambiguity function. It would have been interesting to compare this function to the spectrogram.

One other thing that might have affected my results was the relatively low amount of data samples. Neural networks generally perform well when the number of samples are high so there might be room for improvements there. However, the variability within the results were quite low so it is also quite unlikely that this would have great results. It would also be interesting to see how well these methods generalises to other bird species. The nine species that were selected in this study might not be that representative of birds in general. Another set of birds might have interesting properties in their song which requires higher frequency concentration to detect.

There are other situations where the methods used in this thesis might have worked better. Bird songs were the sole focus of this thesis but there are of course other sound signals that might be interesting to analyse. It is unclear how these results generalises to other sound signals and this would be an interesting research topic.

Finally, my model for selecting the data sequence was a quite crude model. A more sophisticated method would possibly have found a more useful sequence.

6 References

[1] Vielliard J.(2000). *Bird community as an indicator of biodiversity: results from quantitative surveys in Brazil*. Anais da Academia Brasileira de Ciências. vol.72 n.3. DOI:S0001-37652000000300006

[2] Digby A., Towsey M., Bell B., Teal.P (2013) *A practical comparison of manual and autonomousmethods for acoustic monitoring*. Methods in ecology and evolution. Volume 4 issue 7. p 675-683. DOI: 10.1111/2041-210X.12060

[3] Priyadarshani N., Marsland S. Castro I. (2018) *Automated birdsong recognition in complex acoustic environments: a review*. Journal in Avian Biology. Volume49, Issue 5. DOI: 10.1111/jav.01447

[4] Incze A., Janscó H., Szliágyi Z., Farkas A. Sulyok C. (2018) *Bird song recognition using a convolutional neural network.*. Conference: 2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY),Subotica. doi: 10.1109/SISY.2018.8524677

[5] Pál Tóth B., Czeba B. (2016) *Convolutional Neural Networks for Large-Scale Bird Song Classification in Noisy Environment*. Conference: 2016 Conference and Labs of the Evaluation Forum, Évora.

[6] Sankupellay M. Konovalov D. (2018). *Bird Call Recognition using Deep Convolutional Neural Network, ResNet-50* Conference: Proceedings of the Australian Acoustical Society Conference, Adelaide. 134.

[7] Sandsten M. (2020). *Time-Frequency Analysis of Time-Varying Signals and Non-Stationary Processes*. p. 9-12. Compendium.
http://www.maths.lu.se/fileadmin/maths/personal_staff/mariasandsten/TFkompver4.pdf

[8] Sandsten M. (2020). *Time-Frequency Analysis of Time-Varying Signals and Non-Stationary Processes*. p. 21-27. Compendium.
http://www.maths.lu.se/fileadmin/maths/personal_staff/mariasandsten/TFkompver4.pdf

[9] Chen Z, Wu L. (2014) *Blind Source Separation of Dual-Carrier MPPSK Signal Based on Smoothed Pseudo Wigner Distribution.* . 2014 9th International Symposium on Communication Systems, Networks Digital Sign (CSNDSP) DOI: 10.1109/CSNDSP.2014.6923910

[10] Goodfellow I. Bengio Y. Courville A. (2016). *Deep Learning*. MIT Press. p.326

[11] Goodfellow I. Bengio Y. Courville A. (2016). *Deep Learning*. MIT Press. p.164-167

[11] xeno-canto Foundation. *Sharing bird sounds from around the world*[online]. 2020-05-28. https://www.xeno-canto.org/

7 Appendix

7.1 Appendix 1: The settings for the first analysis

The data length of 30 000 had a maximum accuracy of 75%. It was obtained with window length of 512 and 50% overlap. The number of frequency points were 2 048. The network had a convolutional layer with a filter size of 2 and 20 number of filters, a max pooling layer of size 2 with a stride of 2, and a hidden layer with 50 nodes.

For a data length of 20 000 the best accuracy was 73%. The window size overlap and structure of the network was identical to above.

For a data length of 10 000 the best accuracy of 74% was obtained and the window length was 256 with 50% overlap. The number of frequency points were 1024. The structure of the network was identical to the ones above

When the data length was 5 000 an accuracy of 70% was obtained with a window length of 128 and 50% overlap. The number of frequency points were 512. The convolutional layer and max pooling layer were the same as before and there was no hidden layer.

For a data length of 2 500 the maximum accuracy was again 70% with a window length of 32 and 50% overlap. The number of frequency points were 128. The layout of the CNN was identical to the one above.

For a data length of 1 250 the maximum accuracy was 67%. This was obtained with a window length of 32 with 50% overlap and an identical structure of the neural network as above. The number of frequency points were 64.

7.2 Appendix 2: The settings for the second analysis

All the best results for all data length had the same network architecture. The networks had a convolutional layer with a filter size of 2 and 10 number of filters, a max pooling layer of size 2 with a stride of 2, and no hidden layers.

The best result for a data length of 10 000 was 30%. It was obtained with 1 024 frequency points and 1 500 time points. It took 4 hours and 40 minutes to calculate the SPWVD and train the network.

The best result for a data length of 5 000 was 68%. It was obtained with 512 frequency points and 1 700 time points. It took 1 hour and 35 minutes to calculate the SPWVD and train the network.

The best results for a data length of 2 500 was 70%. It was obtained with 512 frequency points and 1 700 time points. It took 1 hour and 36 minutes to calculate the SPWVD and train the network.

The best results for a data length of 1 500 was 69%. It was obtained with 512 frequency points and 1 200 time points. It took 5 hours and 45 minutes to calculate the SPWVD and train the network.

The best results for a data length of 800 was 64%. It was obtained with 512 frequency points and 800 time points. It took 20 minutes to calculate the SPWVD and train the network.

7.3 Appendix 3: The settings for the third analysis

For an initial cut length of 4000 the maximum second cut length that the network could handle was 350. This led to a maximum accuracy of 60%. It took 3 hours and 22 minutes to calculate the distributions and train the network.

The initial cut length of 3500 the maximum second cut length was 300. This led to an accuracy of 63%. It took 4 hours and 35 minutes to calculate the distributions and train the network.

An initial cut length of 2900 gave a maximum accuracy of 61% with the maximum second cut of 280. It took 5 hours and 10 minutes to calculate the distributions and train the network.

With the initial cut length of 2 500 and a maximum second cut of 240 the maximum accuracy achieved was 52%. It took 6 hours and 4 minutes to calculate the distributions and train the network.

7.4 Appendix 4: The settings for the fourth analysis

The spectrogram

All data lengths got the best results from 500 as the pass bound frequency. The networks were pretty much structured in the same way. All networks had had a convolutional layer with a filter size of 2 and 10 number of filters, a max pooling layer of size 2 with a stride of 2. The networks on the data lengths longer than 5 000 had a hidden layer with 50 nodes.

For the data length 30 000 the best results were 79%. This was achieved with a window length of 256 and 50% overlap. The computational time was 40 minutes.

For the data length 20 000 the best results were 73%. This was achieved with a window length of 256 and 50% overlap. The computational time was 30 minutes.

For the data length 10 000 the best results were 74%. This was achieved with a window length of 256 and 50% overlap. The computational time was 16 minutes.

For the data length 5 000 the best results were 74%. This was achieved with a window length of 128 and 50 % overlap. The computational time was 15 minutes.

For the data length 2 500 the best results were 70%. This was achieved with a window length of 32 and 50% overlap. The computational time was 6 minutes.

For the data length 1 500 the best results were 67%. This was achieved with a window length of 32 and 50% overlap. The computational time was 5 minutes.

For the data length 8 00 the best results were 61%. This was achieved with a window length of 32 and 50% overlap. The computational time was 5 minutes.

The Smoothed Pseudo Wigner-Ville

For the data length 10 000, the best accuracy of 30% was achieved with a pass bound frequency of 500. The number of frequency points were set to 512 and the number of time points to 1700. The computational time was 4 hours and 45 minutes.

For the data length 5 000, the best accuracy of 69% was achieved with a pass bound frequency of 300. The number of frequency points were set to 512 and the number of time points to 1500. The computational time was 1 hour and 40.

For the data length 2 500, the best accuracy of 70% was achieved with a pass bound frequency of 500. The number of frequency points were set to 512 and the number of time points to 1700. The computational time was 1 hour and 40 minutes.

For the data length 1 500, the best accuracy of 69% was achieved with a pass bound frequency of 500. The number of frequency points were set to 512 and the number of time points to 1500. The computational time was 6 hours.