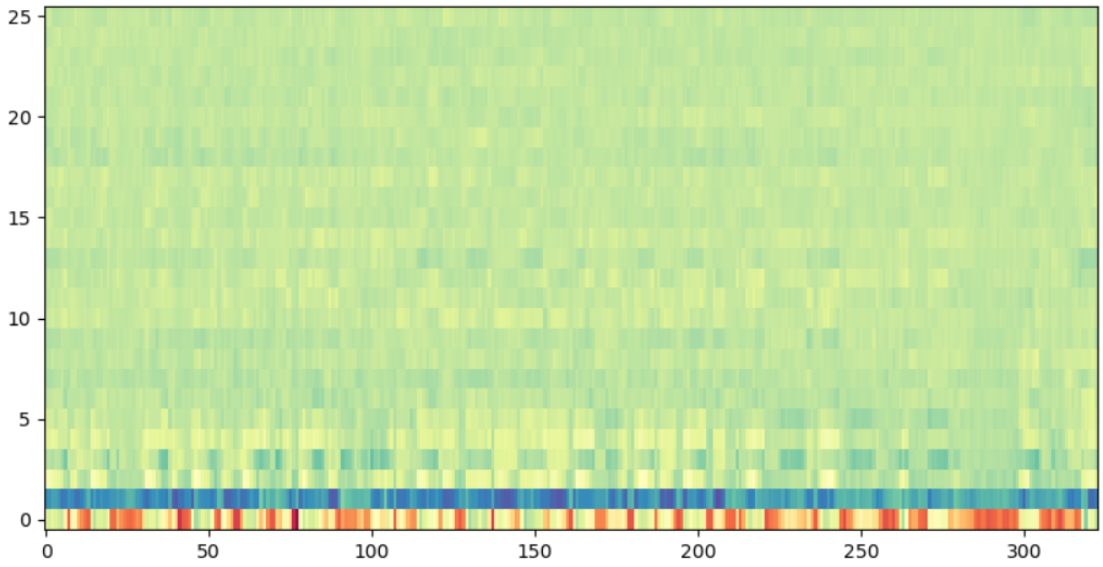


Data Augmentation for Audio Based Machine Learning

Classifying Brachycephalic Obstructive Airway Syndrome (BOAS) in Dogs

Master's thesis in Electrical Engineering

HENRIK PETTERSSON
OLIVIA STENSÖTA

DEPARTMENT OF PHYSICS

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

Master's thesis 2021

Data Augmentation for Audio Based Machine Learning

Classifying Brachycephalic Obstructive Airway Syndrome (BOAS) in Dogs

HENRIK PETTERSSON
OLIVIA STENSÖTA

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Physics
Division of Material Physics
Chalmers University of Technology
Gothenburg, Sweden 2021

Data Augmentation for Audio Based Machine Learning
Classifying Brachycephalic Obstructive Airway Syndrome (BOAS) in Dogs
HENRIK PETTERSSON
OLIVIA STENSÖTA

Supervisors: Magnus Karlsteen, Department of Physics,
Chalmers University of Technology
Maria Dimopoulou, Swedish University of Agricultural Sciences
Ingrid Ljungvall, Swedish University of Agricultural Sciences
Eva Skiöldebrand, Swedish University of Agricultural Sciences

Examiner: Magnus Karlsteen, Department of Physics, Chalmers University of Technology

Master's Thesis 2021
Department of Physics
Division of Material Physics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: An example of an MFCC image of a audio recording.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2021

Data Augmentation for Audio Based Machine Learning
Classifying Brachycephalic Obstructive Airway Syndrome (BOAS) in Dogs
HENRIK PETTERSSON
OLIVIA STENSÖTA
Department of Physics
Chalmers University of Technology

Abstract

Breathing problems of varying degree are common amongst dog breeds with shorter snouts also called brachycephalic dogs. The process of classifying each case consists of a veterinarian visit where tests are preformed to assess the severity on a scale from zero to three. In this master thesis, we aim to simplify this procedure by machine learning and will be working with two hypothesis. Hypothesis I is a continuation of the master thesis *Brachycephalic Obstruction Airway Syndrome (BOAS) classification in dogs based on respiratory noise analysis using machine learning* by Moa Mårtensson. Here we augmented the audio files to generate a larger data set and extracted multiple features. The features include MFCC, ZCR and RMS that are fed to a LSTM network. The second hypothesis aims to classify BOAS(-) and (+), this hypothesis uses frequency data enhanced with SMOTE and a CNN. We show that it is possible to classify BOAS using machine learning, but that more data is required in order to confidently diagnose BOAS. We can conclude that hypothesis II using data collected from the Littmann device shows the best result on unseen audio files. There is a possibility to further develop this into a tool for both veterinarians and dog owners.

This thesis is a collaboration between Chalmers University of Technology and the Swedish University of Agricultural Sciences in Uppsala.

Keywords: machine learning, augmenting, MFCC, RMS, ZCR, SMOTE and BOAS.

Acknowledgements

We would first like to thank our incredible supervisor and examiner Magnus Karlsteen. Without your enthusiasm and help, these months would not have been as joyful as they have nor would our thesis be as exceptional as it is. Thank you!

We would also like to give a special thank you to Moa Mårtensson. Thank you for allowing us to continue your work and for the amazing support you have provided for our entire thesis.

Maria Dimopoulou, Eva Skiöldebrand and Ingrid Ljungvall. Thank you for all your work gathering more data and your quest to vanquishing BOAS. You have also been a great help in our understanding of the breathing problems in dogs.

The computations were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at [SNIC CENTRE] partially funded by the Swedish Research Council through grant agreement no. 2018-05973.

And lastly, but certainty not least, we would like to thank our opponents, Oskar Andersson, Julia Nystrand and Arnita Spule. Thank you for a rewarding discussion as well as valuable feedback on our thesis.

Henrik Pettersson & Olivia Stensöta
Göteborg, June 2021

Contents

List of Figures

List of Tables

List of Abbreviations

BOAS Brachycephalic Obstruction Airway Syndrome.

CNN Convolutional Neural Networks.

FFT Fast Fourier Transform.

JSON JavaScript Object Notation.

LR Learning Rate.
LSTM Long Short-Term Memory.

MFCC Mel-Frequency Cepstral Coefficients.

RMS Root Mean Square.

SMOTE Synthetic Minority Oversampling Technique.

ZCR Zero Crossing Rate.

1

Introduction

The dog was the first animal that humans domesticated. This was because dogs helped warn us against threatening animals, helped us with chores (herding sheep, etc.) but also as company. Today, dogs are used as a pet for the vast majority of people. This has led to a shift in the breeding of the dog, from practical to more aesthetic. In the desire to achieve aesthetic perfection, there have been some complications. Two examples of this are that some dogs have difficulty walking [1] and difficulty breathing [2]. This thesis will focus on detecting dogs who have trouble breathing.

Pugs and bulldogs are dog breeds with a flat face and short nose, also called brachycephalic. Brachycephalic breeds often suffer from Brachycephalic Obstruction Airway Syndrome (BOAS) because of the way their skull is shaped. BOAS can be classified in two ways, with a number between zero and three, where BOAS 0 is a minor inconvenience and BOAS 3 require surgery [3]. The other way is as BOAS(-) and BOAS(+). In layman terms, BOAS means that the dog cannot breathe properly and can, in severe cases, die. By giving potential dog owners the ability to assess if a dog or its offspring may suffer from BOAS before purchase, we can in the long run reduce the number of dogs with BOAS and potentially save future generations from surgery.

According to data from the Swedish Board of Agriculture (Jordbruksverket) [4] there was 12607 French bulldogs in Sweden 2020 compared to 11239 in 2019, an increase of 1368 dogs or 12 %. According to Sveriges television [5], French bulldog is the second most common breed of dog that swedes adopted in 2020. According to a study from 2019 [6], 64% of French bulldogs suffers from at least one of the typical BOAS ailments. This makes it apparent that there is a need to inform the public about the negative aspects of BOAS and eventually produce an easy and accessible way of diagnosing dogs.

This thesis will be a collaboration between Chalmers University of Technology and the Swedish University of Agricultural Sciences in Uppsala. Our thesis is coded in the language Python in Visual Studio Code, central librariews are Librosa[7] and Keras[8]. GitHub was used as edition control as well as a way to collaborate on multiple devices.

1.1 Previous Work

This thesis is the continuation of *Brachycephalic Obstruction Airway Syndrome (BOAS) classification in dogs based on respiratory noise analysis using machine learning* by Moa Mårtensson [9]. In her thesis, two different devices were used to capture recordings. The devices were a Littmann electronic stethoscope and an Olympus linear PCM recorder. Veterinarians captured audio sequences on both devices of dogs breathing, before and after physical exercise.

Machine learning was then used to train a network with the recordings to be able to classify which BOAS class a dog suffered from. The results from the previous thesis are very good with an average classification as high as 88.48 %. The results use the Olympus after physical exercise, data set but has a large variance, the highest accuracy being 92.7 % and the lowest 86.6 %.

1.2 Aim

The aim for this thesis is to make the groundwork for a future application that can classify whether a dog suffers from BOAS by recording its breathing, instead of a thorough examination by a professional. It is also of interest to have an application that veterinarians can use to get an initial indication of the BOAS class. We do this by comparing two different methods to increase the performance of the previous thesis.

1.3 Two Hypotheses

The first hypothesis is to continue Mårtenssons work [9] using the methods she derived as well as introduce new features, use variations of the network and further investigate MFCC settings. We also continue to use four classes during training (BOAS0 - 3). Our second hypothesis explores another approach using a frequency feature and new type of network. For this hypothesis we use two classes (BOAS- and BOAS+). In both hypotheses we will augment our data to generate a larger data set but with two different approaches as will be described more in Chapter 2.

1.4 Limitations

Time will be the main limiting factor both for coding and writing the report as well as the computational time for training networks. We limit ourselves to only implement and test MFCC, RMS, ZCR and frequency as training features. In this thesis we will only tests variants of a LSTM network as well as a two CNN. This thesis will also only investigate two settings when it comes to the pitch, speed and noise augmentations.

2

Theory

In this chapter we discuss the theory behind our thesis. We first present how the data sets have been recorded. We continue with the different methods of augmentation. The next section evolves around the different training features. We conclude with a description of the different networks that are examined in the report as well as an explanation of how we present our results in Chapter 5.

2.1 Audio Recordings

Veterinarians at Uppsala University take in total four different kinds recordings of each dog; two with a Littmann stethoscope [10] and two with a Olympus Dictaphone [11]. One recording on each device is performed on the dog while it is calm and resting while the other is performed after a few minutes of exercise. New recordings are performed continuously throughout the thesis and will hopefully continue after. The files are in `.wav` format and the naming convention is described in Table 2.1.

The Littmann audio files are 30 seconds long while the Olympus files have been edited down to 30-60 seconds. The Olympus Dictaphone has a large scope and captures more than just the patient breathing. We have therefore manually removed disturbances such as talking and doors slamming. The Littmann stethoscope has a more narrow spectrum as it does not pick up surrounding sounds to the same extent as Olympus does. These files, have therefore not been manually edited for disturbances.

Table 2.1: Description of the naming convention for the recording files.

XXX_ABY.WAV

XXX	recordig ID	
AB	Type of recording shorthand	
	LB	Littmann Before
	LA	Littmann After
	OB	Olympus Before
	OA	Olympus After
Y	BOAS grade	0-3

The Littmann and the Olympus files are not identical; the Littmann device has a sampling rate of 4000 samples/s while the Olympus device operate at a more ordinary sample rate of 44100 samples/s.

The dogs are also examined by the veterinarian that classifies a BOAS rating from 0 (best) to 3 (worst). The classification process involves listening to breathing sounds.

2.2 Data Augmentation

The data inherited from the previous thesis consisted of 41 different dogs. The spread of BOAS classes are not uniform as can be seen in Table 2.2.

Table 2.2: The class spread of the original 41 dogs.

BOAS 0	19
BOAS 1	11
BOAS 2	7
BOAS 3	4

Since we have limited data available, data augmentation could be a useful tool to synthesis new training data. There are multiple ways of doing this. We could for instance, either augment the audio files or the images that the network eventually trains on. In hypothesis I, we focus on augmenting the audio files by splitting, time, speed or pitch shifting, and introducing noise. In hypothesis II we augment the frequency data fed to the network using SMOTE.

2.2.1 Splitting

Every audio file can be split into X number of shorter audio segments, which will increase the data files by a factor of X, see Section 2.2.1. This has already been implemented in an earlier stage of the project where each file was split into multiple 3-second-long segments. Splitting will be used on each audio file, after their primary augmentation.

Table 2.3: Very simple table explaining the splitting augmentation. In the original audio file we have one segment containing 9 seconds *(123456789)*. When we split this into 3-second-long segments we get segment 1, 2 and 3.

Original	Segment		
	123456789		
Split	Segment 1	Segment 2	Segment 3
	123	456	789

2.2.2 Time Shift

During time shift, the signal is shifted small steps to the right. When the signal is then split into segments using the splitting augmentation, the first segment will be discarded as it will be padded with zeros and using it would corrupt the data set. Table 2.4 shows a simplified version of this augmentation method. In reality the step size will not be 1 but rather a predetermined number of samples or seconds. The time shift must be shorter than the length of segments after splitting. The total number of segments can be calculated as

$$\text{total segments} = \text{segments} + (\text{segments} - 1) \cdot \frac{\text{segment length}}{\text{time shift}}. \tag{2.1}$$

The variables are thus the number of segments, the segment length and the time the signal will be shifted, the time shift must be larger than zero and smaller than the segment length. It is easy to pick a very small time shift coefficient and gain a lot of extra segments, these new segments will be almost identical to the other augmented segments. Identical training data is not very valuable to train on, but a large time shift coefficient results in few new segments. A balance must be struck between the number of total segments and the difference between them.

Table 2.4: A simplified image of how the time shift and split augmentation would work. The original signal would be shifted until segment 2 is the same as segment 1 was originally.

segment 1	segment 2	segment 3
123	456	789
-12	345	678
–1	234	567
—	123	456

To perform the time shift augmentation, the roll function from `numpy` [12] is utilised and the start is set to 0, see Code 2.1.

```
1  import numpy as np
2  def timeShift(data, sampling_rate, shift):
3      augmented_data = np.roll(data, shift)
4      # Set to silence for heading
5      augmented_data[:shift] = 0
6      return augmented_data
```

Code 2.1: Code for augmentation, timeshift

2.2.3 Pitch Shift

In this process the pitch of the sound is changed while not altering the speed of the file. To augment the pitch we use the librosa pitch shift effects package [13].

2.2.4 Speed Shift

During this procedure the signal is stretched or compressed slightly, however the segment length will be the same when Mel-Frequency Cepstral Coefficients (MFCC)s are extracted. Stretching our data would give more segments, but the individual segments may not contain all the relevant data needed to train the network. On the other hand, compressing our data might result in the individual segments containing more data than necessary, see Table 2.5. The speed shift uses the librosa time stretching package [14].

Table 2.5: A simplified table of how time stretching works. In reality the stretching would be much smaller in relation to the segment than in this simplified table.

	s1	s2	s3	s4	s5	s6	s7	s8	s9
compressed	123	456	789						
original	112	233	445	566	778	899			
stretched	111	222	333	444	555	666	777	888	999

2.2.5 Noise Introduction

To introduce noise would be detrimental if the target network were human, but since the target is a neural network, it can be beneficial. The idea is that the noise will slightly alter the signal, giving more data to train on, but still be similar enough for the original signal to be the main characteristic. We use white noise with zero mean and variance equal to one Code 2.2.

```
1  import numpy as np
2  def noiseAddition(data, noise_factor):
3      noise = np.random.randn(len(data))
4      augmented_data = data + noise_factor * noise
5      # Cast back to same data type
6      augmented_data = augmented_data.astype(type(data[0]))
7      return augmented_data
```

Code 2.2: Code for noise augmentation.

2.2.6 Synthetic Minority Oversampling Technique

As the frequency data is not time dependent like the MFCCs are, we can use a method called SMOTE to up-sample the smaller classes [15]. SMOTE works by finding the neighbor to a data point and then generates extra points in-between them. The augmented result is that all classes have the same size.

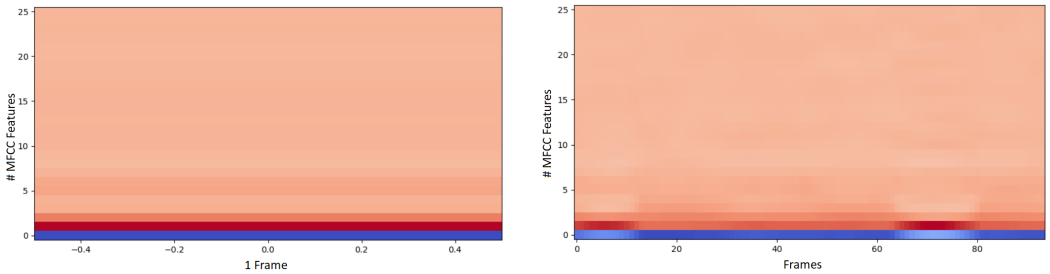
2.3 Training Features

In machine learning, when training a network, there are different features the network can be trained on. Hypothesis I uses MFCC, as this has already been implemented in a previous stage of the project, as well as Root Mean Square (RMS), Zero Crossing Rate (ZCR). We use Librosa [7] which extracts the feature points using the data points inside an n data points wide window. The window is then moved one hop length of m data points. This gives overlapping windows that are eventually stacked into the image. For our second hypothesis we use a frequency feature.

2.3.1 Mel-Frequency Cepstrum

To calculate the MFCC we use the Fourier transform on the audio file. From this we get a Fourier spectrum [16]. On this spectrum we use a logarithmic scale to visualize the magnitude which we then perform a cosine transform on. The resulting spectrum is not in the time or frequency domain since it is a spectrum performed on a frequency spectrum. Because of this, the creators [17] called it the quefrency domain and decided to call this specific spectrum a cepstrum. To extract MFCC we will use the command `librosa.feature.mfcc()` [18].

A segment can look like Figure 2.1b. It consists of frames that are fed into the network. A single frame can look like Figure 2.1a.



(a) One single frame, the frames are strung together to form the segment.

(b) One 3 secound segement.

Figure 2.1: Illustration of MFCC, the x-axis is the index of the frames.

When extracting the MFCC there are 3 mayor parameters; number of MFCC, the Fast Fourier Transform (FFT) window size and the hop length. These can be used to control the amount of data that is used. A higher number of MFCC gives more data and longer run time whereas a smaller hop length gives more data and longer run time. We investigate the optimal settings in Section 3.3.

2.3.2 Root Mean Square

The data points in the window are used to calculate the root mean square value

$$RMS = \sqrt{\frac{1}{n}\sum_{i=0}^{i=n} x_i}. \tag{2.2}$$

The librosa command for this is `librosa.feature.rms()` [19]. This method has been proven to be effective in classifying audio files [20].

2.3.3 Zero Cross Rate

ZCR is a type of feature that can be extracted from audio files. It is represented by a number between 0 and 1, where 0 means that the signal has the same sign and 1 means that the sign changes for every data point. To extract this feature the Librosa command `librosa.feature.zero_crossing_rate()` [21] is used. ZCR has been effectively used to classify audio files [20].

2.3.4 Frequency

When observing the four classes in the frequency spectrum, one can notice a difference between the classes, see Figure 2.2. In attempt to utilize this, we created a method to extract the frequency feature. Unlike the other features, the frequency feature does not depend on time. We can therefore not train a Long Short-Term Memory (LSTM) network. Because of this, we also created a CNN, more about the network in Section 2.4.

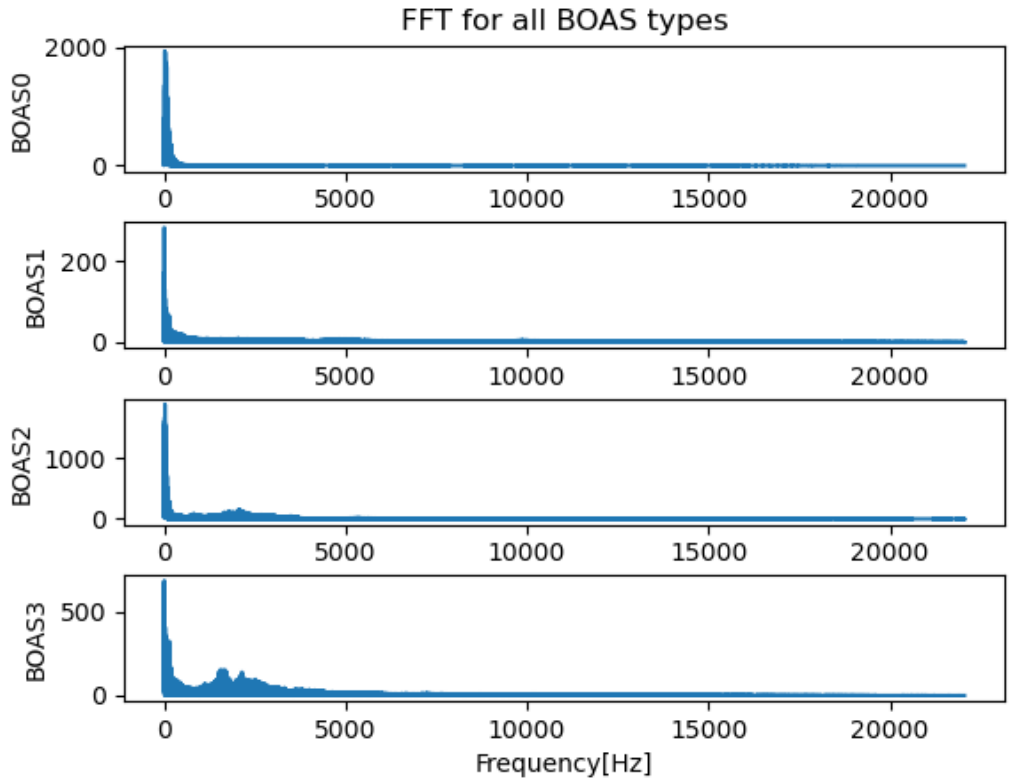


Figure 2.2: FFT plots for Olympus after, one can notice a few differences between the different classes.

2.4 Networks

There are many variables and structures that can be combined to generate a network to the degree that it is not feasible for us to explore them all. For the first hypothesis, we use the network that Mårtensson concluded with, a LSTM network [9] and try different alterations. We will call this network the *base*-(LSTM)-network, see Table 2.6. The variations of this network that are used can be found in Appendix B.

Table 2.6: The base-(LSTM)-network.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 47, 1024)	4358144
dropout (Dropout=0.3)	(None, 47, 1024)	0
lstm_1 (LSTM)	(None, 256)	1311744
dropout_1 (Dropout)	(None, 256)	0
dense (Dense)	(None, 32)	8224
dropout_2 (Dropout=0.3)	(None, 32)	0
dense_1 (Dense)	(None, 4)	132

Because our second hypothesis uses a frequency feature that is not dependent on time, we cannot use a LSTM network. Hence, we construct a CNN that we call the freq network, see Table 2.7. We also try a variation of the network which can be seen in Appendix B.7.

Table 2.7: The freq network.

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 1, 1024)	input dependent
conv1d (Conv1D)	(None, 1, 512)	2621952
dropout (Dropout=0.3)	(None, 1, 512)	0
conv1d (Conv1D)	(None, 1, 128)	327808
dense_1 (Dense)	(None, 4)	516

The models are trained with some additional parameters, such as Learning Rate (LR), batch size and optimizer. These parameters are the same for all networks unless stated otherwise and can be found in Table 2.8.

Optimzer	Adam
Learning rate	10E-5
Batch size	128
Epochs	2500

Table 2.8: The default values when we train the networks.

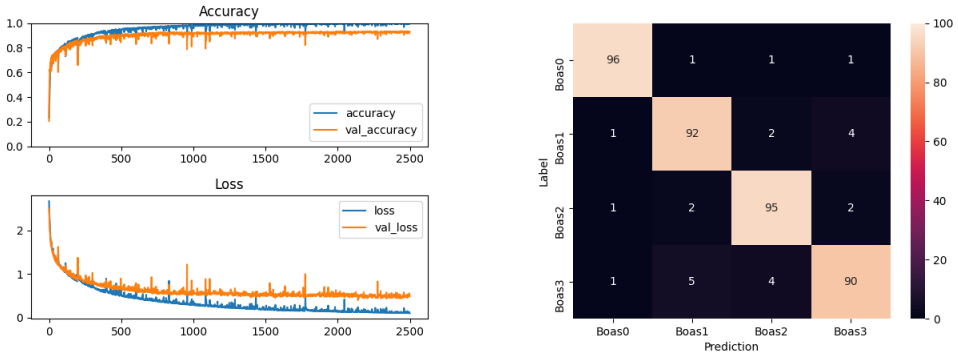
2.5 Presenting the Results

The bulk of the results will be presented in the form of tables that show the accuracy in percentages for the given altered variable, see Table 2.9. We will also present accuracy and loss charts that show how the accuracy and loss changes for each epoch the network is trained. The accuracy is represented by a number between 0 and 1; 0 cannot classify anything correctly and 1 classifies everything correctly. As we have four classes, an accuracy of 25 % would be the same as choosing the right class at random. The model will aim to minimize the loss with each epoch. The blue lines in the charts uses the training data while the orange lines use the validation data i.e. data not used in training, see Figure 2.3a.

The validation data is also represented in the confusion matrix. The confusion matrix has four rows and four columns. Each row represent a batch of segments from the corresponding BOAS class. The numbers represents the percentage of the segments the model classifies as the BOAS class in the rows. This means that the diagonal boxes represent what percentage the model classifies correctly. For example, in Figure 2.3b the final model is correct at classifying 96 % of the BOAS0 segments.

Table 2.9: Example accuracy table.

Altered variable	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
MFCC	73.8	84.4	78.9	10.5



(a) **(b)**

Figure 2.3: Example accuracy and loss chart (a) and confusion matrix (b).

The final way we present our results are with classification tables. In the classification tables, entire audio files that have been omitted from the training are used. The files are split into segments, much like the splitting augmentation, and each segment is classified. The tables show the BOAS class, the filename and what percentage of the audio files segments are classified as the different BOAS classes. For example, in Table 2.10, we see that the model classifies 90 % of the file called *999_lb0.wav* correctly as BOAS0.

Table 2.10: Example classification table.

		Classification [%]			
Class	Filename	BOAS0	BOAS1	BOAS2	BOAS3
BOAS0	999_lb0.wav	90	10	0	0

The examples above are specifically for hypothesis I. The results for hypothesis II will be presented the same way except that it uses two classes instead of four.

3

Methods - Hypothesis I

In the following chapter we demonstrate our approach to data augmentation, settings for extracting features as well as different networks for our first hypothesis. We begin by explaining our training classes. We continue with finding augmentations which increases the accuracy for each recording device. We then move on to finding the optimal settings for extracting MFCC and splitting as well as when to use RMS and ZCR. We end the chapter with an overfitting problem that was caught during the thesis.

3.1 Training Classes

BOAS can be classified in four different categories. When we train a network, we decide what we want the model to classify. Since Mårtensson work [9] showed promising results, we continued to use develop a network that could classify all four grades.

3.2 Augmentation

We augment in two steps; early results indicated a bias towards classifying BOAS0. This was thought to be a result of the larger size of BOAS0 data, see Figure 5.1. Because of this, we first time shift the data so that all four classes contain approximately the same number of segments. We use time shifting for this as it does not alter the signal, it is only sampled in different intervals. We then continue by increasing the amount of data. For this, we use speed and pitch shifting as well as noise introduction.

All tests in this section will be using the Littmann data set as well as the Olympus data set, both recorded after exercise. The base-(LSTM)-network is used and all tests will run for 15 cycles meaning that each setup will have 15 models trained. Statistics from all cycles will then be compiled in the following tables.

The naming convention in Tables 3.2 and 3.3 and later on in the result section is explained in Table 3.1.

Table 3.1: Naming convention in tables; *Even* indicates that the data set has been time shifted to have the same number of audio files in each class. In *Pitch*, the data set has been pitch shifted and in *Speed*, the data set has been speed shifted. In *Noise*, noise has been introduced to the audio file and during *Combo* noise has been introduced as well as pitch and speed shift. And finally, *Combo × 2* means that the combination of all the augmentations has been used with two parameters.

Name	**Variables**
Even	Time shift to get the classes of same size
Pitch	1
Pitch × 2	1 and -1
Speed	1.05
Speed × 2	0.95 and 1.05
Noise	0.005
Noise × 2	0.005 and 0.0075
Combo	Pitch, Speed, Noise
Combo × 2	Pitch × 2, Speed × 2, Noise × 2

For the Littmann data set, pitch and speed shifting increased the accuracy, see Table 3.2. Noise, however, seems to have no or even negative effect. A data set which has been evened as well as pitch and speed shifted shows the most promising results with an average accuracy of 83 % and a variance of 4 %

Table 3.2: The table shows the minimum, maximum and average accuracy of the network when introducing noise, speed and pitch shifting. The settings described in Section 3.3.1 are used, except for the previous results [9] which used a hop length of 512.

Littmann	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
Previous results	52.4	69.5	61.8	17.1
Even segments	68.8	84.9	78.7	16.1
Even segments + pitch	75.3	85.1	81	9.8
Even segments + speed	81	85.7	83.1	4.7
Even segments + noise	68.1	77	72.7	8.9
Even segments combo × 2	74.3	78.7	76.8	4.4
Even segments, pitch, speed	81	85.1	83	4

The results for the Olympus data set are different and can be seen in table 3.3. We see that we had an increase in accuracy and a decrease in variance for both speed shift and noise introduction separately but also combined. Shifting the pitch resulted in a smaller variance but a lower overall average. With all augmentations, the variance was decreased from almost 10 % to $\approx$ 5 %. The highest average accuracy was obtained by combining all augmentations with two parameters.

Table 3.3: The minimum, maximum and average accuracy of the network when introducing noise, speed and pitch shifting. The settings described in Section 3.3.2 are used.

Olympus	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
Previous results	81.3	94.3	86.9	13
Even segments	85.9	95.1	91.6	9.2
Even segments + pitch	87.5	93.6	91.2	6.1
Even segments + speed	91.3	96.7	94.4	5.4
Even segments + noise	91.6	96.8	94.1	5.2
Even segments + speed + noise	92.7	98.3	96.2	5.6
Even segments + combo × 2	97.2	98.8	97.7	1.6

Comparing Table 3.3 and Table 3.2 it is apparent that the data sets react differently to the augmentations. For the Olympus data set, noise augmentation showed the greatest improvement while for the Littmann data set, noise was by far the worst type of augmentation. Because of this we will use all three types of augmentation for the Olympus data sets but only pitch and speed shifting for the Littmann data sets.

3.3 Optimal Settings

To achieve the best results, the optimal values for the different parameters used to extract MFCC has to be determined. It is also of interest to see if features such as RMS and ZCR show improving results. For all tests, the base-(LSTM)-network is used. The tests will run for 15 cycles, statistics from all cycles will then be compiled in all tables.

3.3.1 Littmann Settings

The default variables used to extract the json files from the Littmann data sets are

#MFCC	39
Windowsize	2048
hop length	256

Tables 3.4 and 3.5 suggests that a hop length of 128 and 26 MFCC should yield the best result. The results in Table 3.6 is not as decisive as it improves with both a higher and a lower FFT window.

Table 3.4: The table shows the minimum, maximum and average accuracy of the network with different hop lengths; default = 512, 256, 128 and 64 for the Littmann data set, recorded after exercise, with even number of segments in each BOAS class.

Hop length	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
512	72.8	83.4	77.9	5.6
256	68.8	84.9	78.7	16.1
128	80.4	86.9	83.2	6.5
64	72.8	82.9	78.2	10

Table 3.5: The table shows the minimum, maximum and average accuracy of the network with different number of MFCC; 39, 26 and 13 for the Littmann data set, recorded after exercise.

MFCC	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
39	68.8	84.9	78.7	16.1
26	72.8	84.4	78.8	11.6
13	68.8	82.9	74.1	14.1

Table 3.6: The table shows the minimum, maximum and average accuracy of the network with different width of FFT window length; 1024, 2048 and 4096 for the Littmann data set, recorded after exercise.

FFT window	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
1024	73.8	84.4	78.9	10.5
2048	68.8	84.9	78.7	16.1
4096	72.8	81.4	76.8	8.5

From Table 3.7 it seems that RMS can be used to increase the accuracy to a lower degree compared to MFCC. RMS has a single row of data, which can be compared to our default case of MFCC that has 39 rows of data. Because of this, the lower accuracy is not surprising. ZCR is around 25 % which is the same as randomly choosing a class.

Table 3.7: The table shows the minimum, maximum and average accuracy of the network with additional training features for the Littmann data set, recorded after exercise.

Feature	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
MFCC	68.8	84.9	78.7	16.1
RMS	26.1	38.1	32.7	12
ZCR	19.5	29.6	26.3	10
MFCC+RMS+ZCR	74.8	84.4	79.4	9.5
MFCC+RMS	75.3	84.9	78.9	9.5

3.3.2 Olympus Settings

The default variables used to extract the JavaScript Object Notation (JSON) files from the Olympus data sets are

#MFCC	39
Windowsize	2048
hop length	512

All tests are preformed with the Olympus data set recorded after exercise which has been augmented to have the same number of segments in every BOAS class.

The default settings which have been used in previous tests for the Olympus data sets are MFCC = 39, FFT window = 2048 and hop length = 512. In Table 3.8 we can see that a higher hop length gives a higher accuracy on average, however a hop length of 512 gives the highest maximum accuracy. Table 3.9 suggests that 26 coefficients are preferable as it produces a high accuracy with smaller variance than for example 39 coefficients. The width of the FFT window should be 1024 according to Table 3.10.

Table 3.8: The table shows the minimum, maximum and average accuracy of the network with different hop lengths; 512, 256 and 128 for the Olympus data set, recorded after exercise which has been augmented to have the same number of segments in every BOAS class.

Hop length	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
1024	87.2	93.6	92.2	6.4
512	85.9	95.1	91.6	9.2
256	85.7	93.6	90.4	7.8

Table 3.9: The minimum, maximum and average accuracy of the network with different number of MFCC; 39, 26 and 13 for the Olympus after exercise data set which has been augmented to have the same number of segments in every BOAS class.

MFCC	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
39	85.9	95.1	91.6	9.2
26	87.7	94.6	91.3	6.8
13	86.7	92.6	89.6	5.9

Table 3.10: The minimum, maximum and average accuracy of the network with different width of FFT window length; 1024, 2048 and 4096 for the Olympus after exercise data set which has been augmented to have the same number of segments in every BOAS class.

FFT window	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
1024	91.6	95	93.2	3.4
2048	85.9	95.1	91.6	9.2
4096	89.7	95.5	92.6	5.8

In Table 3.11 we notice that both RMS and ZCR perform just slightly better than random choosing a class, but when all three are combined the result are much better than if only using MFCC. The minimum accuracy has been raised by $\approx 5\,\%$ and the variance is halved.

Table 3.11: The minimum, maximum and average accuracy of the network with additional training features for the Olympus data set recorded after exercise.

Feature	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
MFCC	85.9	95.1	91.6	9.2
RMS	21.5	35.7	26.0	14
ZCR	25	41.6	29	16
MFCC+RMS+ZCR	90.6	95	92.6	4.4
MFCC+RMS	86.2	94.6	91.7	8.3

3.3.3 Settings for Splitting

As mentioned before, splitting has been used even before this thesis. Each audio file was split into different segments of three seconds. The width of the image that was created and fed to the network is calculated by

$$\frac{\text{segment length} \cdot \text{sample rate}}{\text{hop length}}. \tag{3.1}$$

We know that the sample rate for Littmann is 4000 samples/s and 44100 samples/s for Olympus, that means that Olympus has 11 times more data than Littmann. If we use the inherited hop length of 512 on both Littmann and Olympus we get 23 and 258 columns respectively to feed the LSTM network. The images created from the Littmann recordings are significantly smaller than those from Olympus. Different segment lengths were therefore tested on the Littmann data set recorded after exercise using the optimal settings found in Table 5.4. The tests run for 15 cycles and statistics from all cycles will then be compiled in all tables.

In Table 3.12 we see that a segment length of 5 seconds performs better in every aspect except for the variance. It would therefore be preferable to use a 5 second

segment length for the Littmann data set. These tests were performed in the final stages of the thesis which means that the tests performed in the result chapter still use a 3-second-segment length.

Table 3.12: The minimum, maximum and average accuracy of the network with additional training features for the Littmann data set recorded after exercise using optimal settings.

Segment length	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
3 sec	83.2	87.1	85.2	3.8
5 sec	83.6	90.1	87.8	6.5

3.4 Overfitting Problem

In this section we touch upon an overfitting complication that is seen in Chapter 5 and discussed in Chapter 7. Since discovering this obstacle, attempts have been implemented to circumvent the problem. Here we account for the two methods used and discuss their potential.

When comparing the results we got during training with the results we got during classification of omitted files, we get very different results. Instead of correctly classifying up towards 90 % of the segments, the models classifies almost everything as BOAS0. Earlier in the project we got a bias towards BOAS0 in the training as well, we tried to counter this by evening the boas classes, see Section 3.2. This had positive effects on the training; both the training accuracy and the validation confusion matrices showed significant improvement. But when testing the models on the omitted files the results were very unsatisfactory as there was an aversion to BOAS 3; the class that the model was most apt on identifying during training. We have successfully created a hidden over-fitting problem.

We make two different attempts to solve this; we limit the data in the classes to the size of the smallest class (BOAS3) and we use an unevened data set with class-weights.

3.4.1 BOAS3 Size Limit

In Table 3.13 we see that the smaller data set has similar training results compared to a larger data set. During classification, Table 3.14, the network has a bias towards BOAS 0 and 1. This is not better nor worse than other results. Thus this seems to be a dead end and will be abandoned.

Table 3.13: The minimum, maximum and average accuracy of the network. The first network uses the Olympus data set recorded before exercise which has been evened with optimal settings. The network ran 3 cycles. The networks uses a smaller version of the Olympus before exercise data set but with only 13 dogs. Optimal settings were used and the network ran for two cycles.

Segment length	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
Optimal settings	88.7	90	89.4	1.2
Smaller data set	89.3	90.6	89.9	1.2

Table 3.14: Classification on omitted files. The network uses a smaller version of the Olympus before exercise data set but with only 13 dogs. Optimal settings were used and the network ran for two cycles.

		Classification [%]			
Class	Filename	BOAS0	BOAS1	BOAS2	BOAS3
BOAS1	010_ob1.wav	0	70	0	30
BOAS0	035_ob0.wav	40	50	0	10
BOAS1	074_ob1.wav	83	16	0	0
BOAS3	076_ob3.wav	37	45	16	0
BOAS0	081_ob0.wav	64	35	0	0
BOAS2	106_ob2.wav	14	85	0	0

3.4.2 Class-weights

Another attempt to counter the hidden overfitting problem is to use class weights. In this approach, the networks verdict is influenced by the weight of each class. To test this method, the Olympus data set, recorded before exercise, which has not been evened is used. When training the model it was fed the class-weight vector

$$\{0 : 1/49, 1 : 1/35, 2 : 1/26, 3 : 1/13\}.$$

This means that a BOAS0 file is worth less than the other classes during training. As the class-weight decreases the impact of each segment.

Table 3.15: The minimum, maximum and average accuracy of the network. The networks uses the Olympus data set, before exercise. Optimal settings were used, except for the evened time shift. OBS, not all runs are added here, only on run was finished for the optimal.

Segment length	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
Default settings	74.8	80.6	77.3	5.8
Optimal settings	86.4	86.4	86.4	0
Optimal settings, higher LR	85.4	89.9	88.6	4.4

Table 3.16: Classification on omitted files. The network used is the best performing in Table 3.15. Optimal settings were used, except for the evened time shift. The network ran for one cycle.

		Classification [%]			
Class	Filename	BOAS0	BOAS1	BOAS2	BOAS3
BOAS1	010_ob1.wav	40	40	10	10
BOAS0	035_ob0.wav	90	10	0	0
BOAS1	074_ob1.wav	100	0	0	0
BOAS3	076_ob3.wav	100	0	0	0
BOAS0	081_ob0.wav	78	21	0	0
BOAS2	106_ob2.wav	64	35	0	0

The results in Table 3.15 for *Default settings* and *Optimal settings* shows a decrease in accuracy compared to the result presented in Figure 5.5a. However, the accuracy plot in Figure 3.1 we notice that both the training and validation curves are still increasing after 2500 epochs. Therefor the learning rate is increased in *Optimal settings, higher LR* from $1E-5$ to $1E-4$. But as seen in Table 3.16 this did not improve the classification.

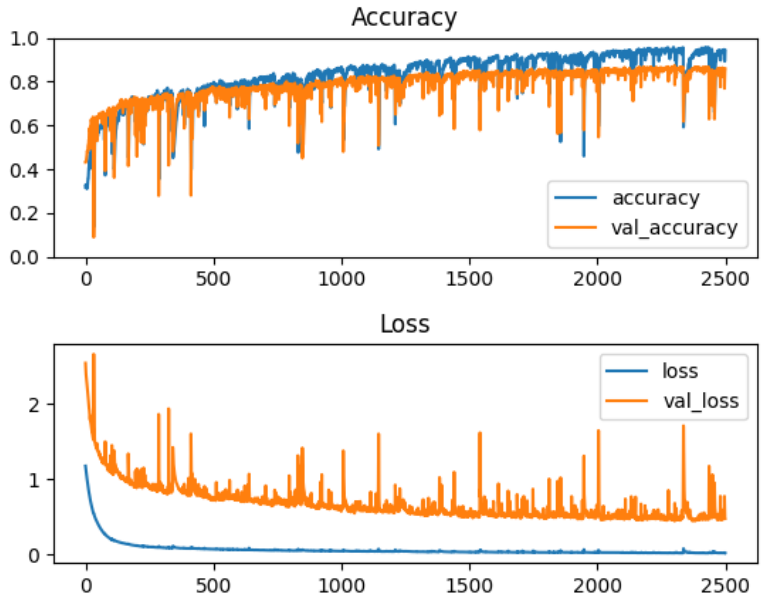


Figure 3.1: In the accuracy we can see that both the training and the validation is still increasing after the 2500 epochs.

4

Methods - Hypothesis II

In this chapter we explore our methods for hypothesis II. First we talk about the number of training classes. We then find the best network and augmentation settings. We conclude with finding the optimal settings for the frequency feature.

4.1 Training Classes

As we have mentioned multiple times before, BOAS has four grading levels. In our first hypothesis, we use the four BOAS grades as our classes. In our second hypothesis, we use the fact that both grade 0 and 1 indicates a somewhat normal breathing that does not require surgery. While grade 2 and 3 indicates severe breathing problems and does require surgery. Instead of developing a network that can distinguish between all four categories, we crate a model that classifies BOAS- and BOAS+.

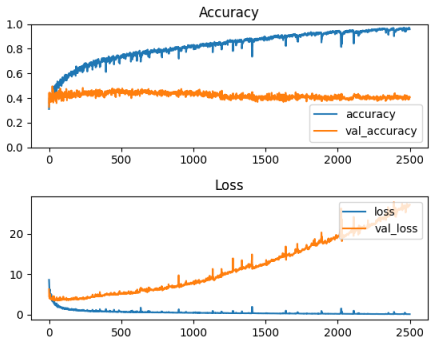
4.2 Networks and Data Augmentation

For this hypothesis we use the frequency feature and augment our data using SMOTE. In SMOTE, one can choose the number of neighbors that is used to create the new data points. The following tests are to establish which of the two networks is favorable as well as what the number of neighbors should be. All tests use the the Olympus data set, recorded before exercise. The tests run for 15 cycles each. To save time and computing resources we do not run these tests for all four data sets.

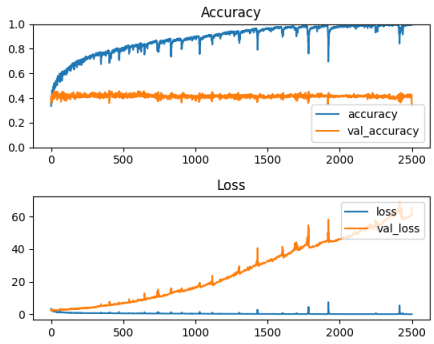
When comparing the two networks freq and freq2, in Table 4.1, it is hard to draw any real conclusion. If we look at the accuracy plots in Figure 4.1, we see that the training accuracy for freq2 approaches 1 much faster than freq. Thus, the freq2 network is considered to be superior. Both networks show signs of overfitting as can be seen in the loss charts in Figure 4.1. This is believed to be because of the class spread of the dogs. To counter this we try to augment the data using SMOTE to even out the classes. As we can see in Table 4.2 this proves to have a significant effect. The choice of the number of neighbors seems to have little effect, although a higher number seems to have a lower variance.

Table 4.1: The minimum, maximum and average accuracy as well as the variance for two different CNN.

Network	Min [%]	Max [%]	Average [%]	Variance [%]
freq	43.7	45.3	44.5	1.5
freq2	39.5	47.2	42.9	7.6



(a) Accuracy/loss for the freq network.



(b) Accuracy/loss for the freq2 network.

Figure 4.1: The accuracy approaches 1 faster for the freq2 network than for the freq network.

Table 4.2: The minimum, maximum and average accuracy as well as the variance for the freq2 network. We vary the neighbors while using SMOTE with 2, 5 and 9 neighbors.

SMOTE	Min [%]	Max [%]	Average [%]	Variance [%]
2	66.2	71.4	68.9	5.1
5	66.9	72.5	69.6	5.6
9	67.7	70.2	69	2.4

4.3 Optimal Settings

When taking the FFT of our signal, we get an array which will be of different size for Littmann and Olympus. For our frequency feature, we then have to decide how many of the data points to use. A rough estimate is to use half the sampling rate (2000 for Littmann and 22050 for Olympus) as the signal will repeat itself after that. All tests use the the Olympus and Littmann data sets, recorded before exercise with the freq2 network. The tests run for 15 cycles each except for Olympus with 22050. Because of the large number of trainable parameters, we only run it for one cycles.

Table 4.3: The minimum, maximum and average accuracy of the network, with different settings for the Littmann data set, recorded before exercise.

# data points	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
1500	50.3	80.2	75.7	29
2000	50.4	79.4	75.7	28

In Table 4.3 the results are pretty much interchangeable. As fewer data points gives a faster training, 1500 data points were chosen.

Table 4.4 shows that a higher range of data points is preferable. Since Olympus has a higher sampling rate, this is expected. There are however limits to our computing capabilities, we therefore use 22050 data points for the Olympus data sets.

Table 4.4: The minimum, maximum and average accuracy of the network, with different settings for the Olympus data set, recorded before exercise. The 22050 run is only ran once because of its size.

# data points	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
1500	50.3	79.8	73.3	29
3000	73.4	79.6	76.4	6.2
22050	82.4	82.4	82.4	0

5

Results - Hypothesis I

In the following chapter we will present the final results for hypothesis I. We begin with the data sets that have been augmented in various ways with different settings to determine the best data sets. The we continue with how the networks created with the best performing data sets performs with new recordings. We end this chapter with comparing the different networks.

5.1 Augmentation

Originally the data sets consisted of 41 dogs. During the project, 83 additional dogs have been recorded totaling in 124 dogs. The interest of BOAS classification amongst dog owners is great so even more recordings are expected to take place. From this alone, the data set has increased more than 300 % during the project. The data regarding the Olympus data set recorded after training, is first time shifted to make the BOAS classes more equally large. This gives ≈ 230 recordings. These 230 recordings are then augmented by pitch, speed and noise according to the combination in table 3.1. The end result is the equivalent of 1610 recordings, which is an increase by ≈ 1300 % or 39 times the data that the previous thesis had to train with. For the Littmann data sets, the equivalent is about 1150 which corresponds to about 920 %.

The final distribution of BOAS classes can be seen in Figure 5.1 and the breed of dogs in Figure 5.2.

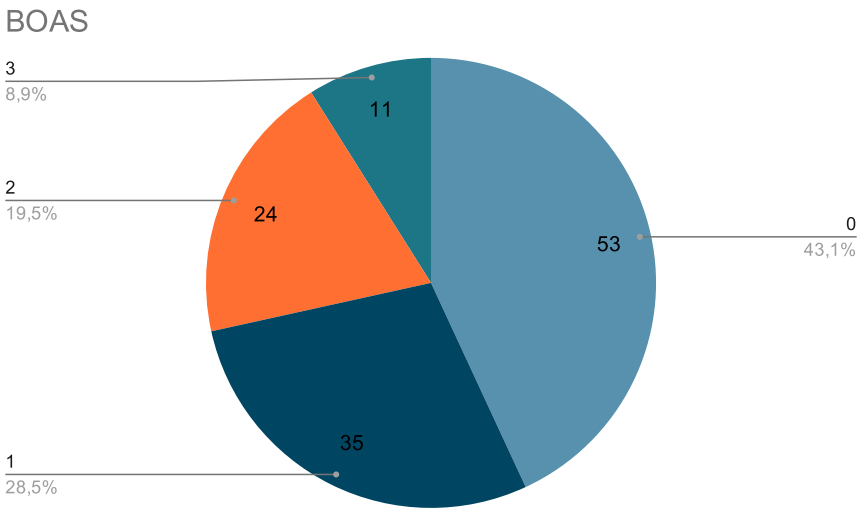


Figure 5.1: The distribution of the different BOAS classes in the recorded data.

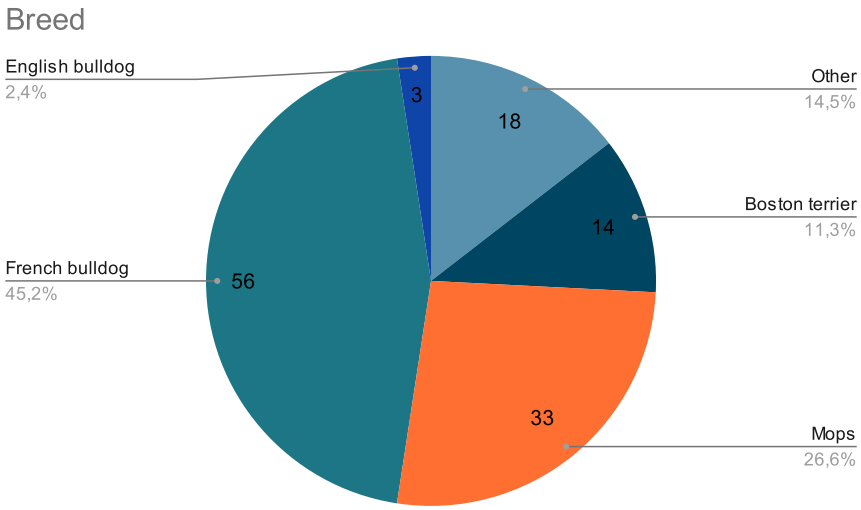


Figure 5.2: The distribution of the different dog breeds in the recorded data. breeds with two or fewer dogs are placed in the *other* section.

All tests are performed with the base-(LSTM)-network. The tests will run for 15 cycles and statistics will then be compiled in tables.

5.1.1 Littmann Data Set

The optimal values were achieved by various tests as explained in Section 3.3.1 and can be seen in Table 5.1. In Table 5.2 we see that the Littman data set, recorded before exercise with optimal settings performs the best in all categories with an average accuracy of 86.1 %. The same can be said for the Littmann data

set recorded after exercise also using the optimal settings. Here, we have an average accuracy of 85.2 %, see Table 5.3.

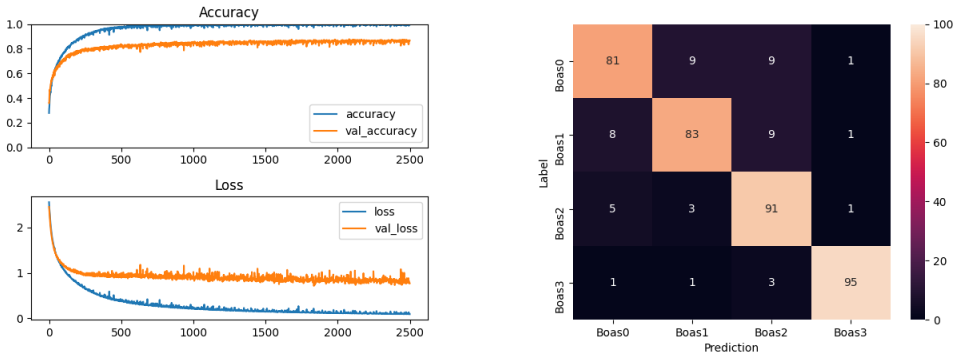
An accuracy/loss chart as well as a confusion matrix for the best performing models for both Littmann data sets can be seen in Figures 5.3 and 5.4. We see that the training data is close to perfect while the loss could be lowered further. It would seem that the Littmann data set, recorded before exercise, is better at classifying BOAS3 compared to the Littmann data set, recorded after exercise, based on the confusion matrices.

Table 5.1: The optimal values for each variable for the Littmann data set before and after exercise based on Section 3.3.1

Variable	**Values**
Timeshift	Even
Pitch	-1, 1
Speed	0.95, 1.05
#MFCC	26
FFT window	2048
Hop length	128
Network	base-(LSTM)-network
Additional features	RMS

Table 5.2: The minimum, maximum and average accuracy of the network with different settings and augmentations for the Littmann data set, recorded before exercise. *Previous results* are the results from the previous thesis [9].

Littmann Before	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
Previous results	69.5	77.1	66.4	18.1
Default settings	63	69.2	66.5	6.1
Even, default settings	63.3	70	65.5	6.6
Optimal settings	83.4	87.4	86.1	3.9

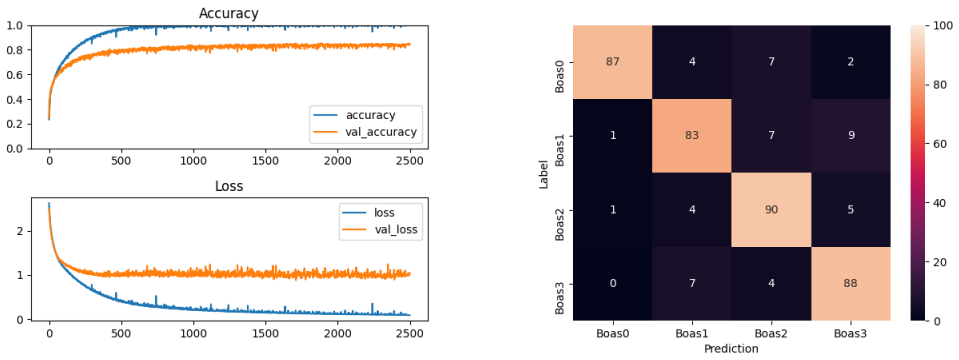


(a) **(b)**

Figure 5.3: Training and validation (a) as well as confusion matrix (b) for the Littmann before exercise data set using the optimal settings.

Table 5.3: The minimum, maximum and average accuracy of the network with different settings and augmentations for the Littmann data set, recorded after exercise. This training uses all data acquired before 2021-05-01. *Previous results* are the results from the previous thesis [9].

Littmann After	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
Previous results	52.4	69.5	61.8	17.1
Default settings	55.3	64	58.9	8.6
Even, default settings	61.6	67.1	64.2	5.4
Optimal settings	83.2	87.1	85.2	3.8



(a) **(b)**

Figure 5.4: Training and validation (a) as well as confusion matrix (b) for the Littmann after exercise data set using the optimal settings.

5.1.2 Olympus Data Set

In Section 3.3.2, tests were performed to assess the optimal values for different parameters and the result can be seen in Table 5.4. In Table 5.5 we see that the

data set which uses the optimal settings for the Olympus data set, recorded before exercise, has the highest performance out of the data sets. They yield an average accuracy of 89.4 % and a variance of only 1.2 %. In contrast to this we see that in Table 5.6 we have a maximum accuracy of 93.6 %, however, the accuracy presented here is a compilation of only one run.

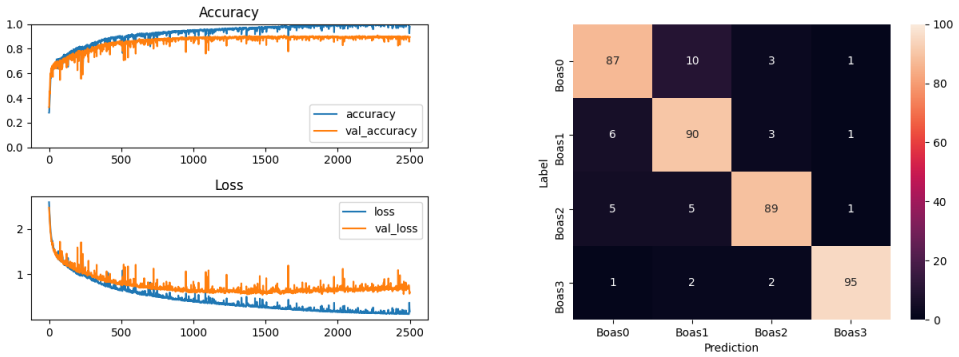
An accuracy/loss chart as well as a confusion matrix for the best performing models for both Olympus data sets can be seen in Figures 5.5 and 5.6. We see that both the Olympus data sets perform better for both accuracy and loss than the Littmann data sets. The Olympus data set, recorded after exercise, has a better performance for all classes except for BOAS 3.

Table 5.4: The optimal values for each variable for the Olympus data set based on Section 3.3.2

Variable	**Value**
Timeshift	Even
Pitch	-1, 1
Speed	0.95, 1.05
Noise	0.005, 0.0075
#MFCC	26
FFT window	2048
Hop length	256
Network	base-(LSTM)-network
Aditional feature	RMS, ZCR

Table 5.5: The minimum, maximum and average accuracy of the network with different settings and augmentations for the Olympus data set, recorded before exercise. This training uses all data acquired before 2021-05-01. *Previous results* are the results from the previous thesis [9].

Olympus Before	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
Previous results	81.4	92.4	86.3	11
Default settings	73.2	83.5	80	10
Even, default settings	85.3	89.1	87.2	3.8
Optimal settings	88.7	90	89.4	1.2

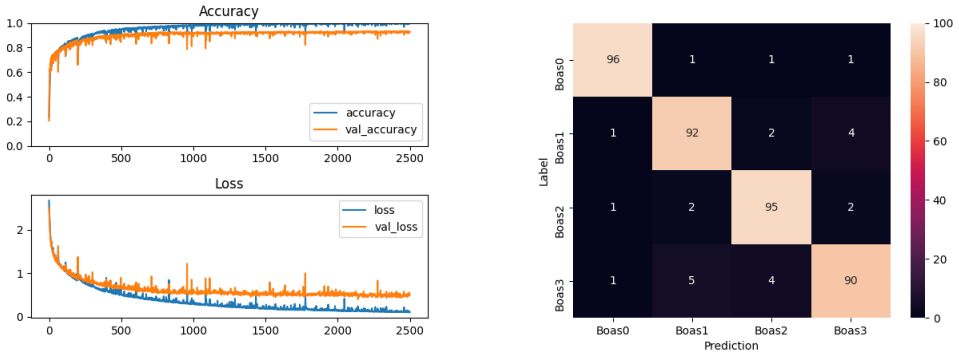


(a) **(b)**

Figure 5.5: Training and validation (a) as well as confusion matrix (b) for the Olympus before exercise data set using the optimal settings.

Table 5.6: The minimum, maximum and average accuracy of the network with different settings and augmentations for the Olympus data set, recorded after exercise. This training uses all data acquired before 2021-05-01. *Previous results* are the results from the previous thesis [9].

Olympus After	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
Previous results	81.3	97.3	86.9	13
Default settings	74.3	82.9	79.3	8.6
Even, default settings	87.5	90	88.7	2.5
Optimal settings	93.6	93.6	93.6	0



(a) **(b)**

Figure 5.6: Training and validation (a) as well as confusion matrix (b) for the Olympus after exercise data set using the optimal settings.

5.2 Classification

To test the final models, we use files that we randomly omitted from the training. Note, there is only one file each of BOAS 2 and BOAS 3 because there are much fewer recordings in these classes and it might damage the training by omitting more. The audio files are split into 3-seconds-segments which the model will try to classify.

5.2.1 Littmann Data Set

The classifications is mostly incorrect for the Littmann data set, recorded before exercise with optimal settings. The result are leaning towards favoring BOAS0, see Table 5.7. The network can only classify two out of six dogs correctly. And even though it is very certain of these two classifications, it is also certain of other classifications which are incorrect.

Table 5.7: Classification on omitted files for the Littmann data set, recorded before exercise, using optimal settings.

		Classification [%]				
Class	Filename	BOAS0	BOAS1	BOAS2	BOAS3	Correct
BOAS1	013_lb1.wav	90	0	10	0	X
BOAS0	033_lb0.wav	90	0	10	0	✓
BOAS2	060_lb2.wav	27	18	0	54	X
BOAS3	068_lb3.wav	81	0	9	9	X
BOAS1	093_lb1.wav	58	0	0	41	X
BOAS0	102_lb0.wav	91	0	8	0	✓

In Table 5.8, the Littmann data set recorded after exercise with optimal settings, there is a bias towards BOAS3. It does classify BOAS2 is correctly, which is rare. Overall, the results are poor as it can only classify one out six dogs correctly.

Table 5.8: Classification on omitted files for the Littmann data set, recorded after exercise, using optimal settings.

		Classification [%]				
Class	Filename	BOAS0	BOAS1	BOAS2	BOAS3	Correct
BOAS0	006_la0.wav	0	0	0	100	X
BOAS1	014_la1.wav	0	40	0	60	X
BOAS2	055_la2.wav	0	0	72	27	✓
BOAS3	069_la3.wav	0	8	58	33	X
BOAS1	093_la1.wav	0	0	0	100	X
BOAS1	102_la0.wav	0	0	30	69	X

5.2.2 Olympus Data Set

In contrast to the very high performance in Table 5.5, we see that when we use our model, created with the Olympus recordings before exercise data set, our classification is mostly incorrect, see Table 5.9. We only classify two out of six dogs correctly. We seem to have a hard time classifying BOAS2 and 3 as nearly no segments are classified as BOAS3.

Table 5.9: Classification on omitted files for the Olympus data set, recorded before exercise, using optimal settings.

		Classification [%]				
Class	Filename	BOAS0	BOAS1	BOAS2	BOAS3	Correct
BOAS1	010_ob1.wav	50	10	30	10	X
BOAS0	035_ob0.wav	90	10	0	0	✓
BOAS1	074_ob1.wav	100	0	0	0	X
BOAS3	076_ob3.wav	4	95	0	0	X
BOAS0	081_ob0.wav	78	21	0	0	✓
BOAS2	106_ob2.wav	50	42	7	0	X

We see similar results for the Olympus recordings after exercise seen in Table 5.10. However, this model has an easier time classifying BOAS 2 and 3 although the classification is mostly incorrect.

Table 5.10: Classification on omitted files for the Olympus data set, recorded after exercise, using optimal settings.

		Classification [%]				
Class	Filename	BOAS0	BOAS1	BOAS2	BOAS3	Correct
BOAS1	029_oa1.wav	30	0	40	30	X
BOAS0	038_oa0.wav	0	10	10	80	X
BOAS3	059_oa3.wav	83	8	0	8	X
BOAS2	063_oa2.wav	16	24	60	0	✓
BOAS1	090_oa1.wav	7	30	23	38	X
BOAS0	103_oa0.wav	0	0	14	85	X

5.3 Networks

As mentioned before, all previous tests have used the base-(LSTM)-network found in Table 2.6. We further developed that network to try to increase the accuracy. All tests in this section use the Littmann data set, recorded after exercise, or the Olympus data set, recorded after exercise, which both have been evened. The default settings found in Section 3.3.1 and Section 3.3.2 are used and each network has been run 15 times.

We see that the base-(LSTM)-network as well as network 4 has the highest average accuracy. However, the base-(LSTM)-network has the lowest variance of the two, see Table 5.11.

In Table 5.12 we can see that the base-(LSTM)-network as well as network 5 and 6 has the highest average accuracy. Network 6, which is a very small network has a high variance while the base-(LSTM)-network and network 5 both have a lower variance of 4.3 %.

Table 5.11: The minimum, maximum and average accuracy of the network with different networks with the Littmann data set recorded after exercise.

Network	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
Base	62.1	67.3	64.5	5.1
2	61.3	66.8	64.2	5.4
3	59.8	66.7	64.0	6.9
4	59.8	67.0	64.5	7.2
5	59.3	67.3	63.6	8.0
6	53.8	59.4	56.9	5.6
7	60.1	64.8	62.3	4.7

Table 5.12: The minimum, maximum and average accuracy of the network with different networks with the Olympus data set recorded after exercise.

Network	**Min** [%]	**Max** [%]	**Average** [%]	**Variance** [%]
Base	85.6	89.9	88.3	4.3
2	82.7	89.9	86.7	7.1
3	83.9	90.2	86.6	6.3
4	83.9	89.3	87.8	5.4
5	86.6	90.9	88.3	4.3
6	84.0	91.5	88.3	7.4
7	79.1	85.7	82.3	6.6

6

Results - Hypothesis II

In this chapter we show the classification results for each data set, with optimal settings, using our second hypothesis. Each of the Littmann tests have run for 15 cycles while the Olympus tests only ran for 1 due to the large number of parameters.

In Table 6.1 we can see that the model classifies five out of six dogs correctly. It is unsure about the last dog classifying 50 % of the segments as BOAS- and 50 % as BOAS+. The results are of the data set Littmann, recorded before exercise.

Table 6.1: Classification on omitted files for the Littmann data set, recorded before exercise, using optimal settings.

		Classification [%]		
Class	Filename	BOAS-	BOAS+	Correct
BOAS1	013_lb1.wav	100	0	✓
BOAS0	033_lb0.wav	60	40	✓
BOAS2	060_lb2.wav	18	81	✓
BOAS3	068_lb3.wav	36	63	✓
BOAS1	093_lb1.wav	50	50	-
BOAS0	114_lb0.wav	30	69	✓

For the Littmann data set recorded after exercise, the model is correct at classifying four out of six, see Table 6.2.

Table 6.2: Classification on omitted files for the Littmann data set, recorded after exercise, using optimal settings.

		Classification [%]		
Class	Filename	BOAS-	BOAS+	Correct
BOAS0	006_la0.wav	60	40	✓
BOAS1	014_la1.wav	50	50	-
BOAS2	055_la2.wav	72	27	X
BOAS3	069_la3.wav	41	58	✓
BOAS1	093_lb1.wav	75	25	✓
BOAS0	102_la0.wav	92	7	✓

table 6.3 shows our results for the Olympus data set, recorded before exercise. We see that the model cannot classify a single dog correctly. We see better results in Table 6.4, where the model can correctly classify four out of six dogs.

Table 6.3: Classification on omitted files for the Olympus data set, recorded before exercise, 22050 frequency points. Very bad result

		Classification [%]		
Class	Filename	BOAS-	BOAS+	Correct
BOAS1	010_ob1.wav	40	60	X
BOAS0	035_ob0.wav	20	80	X
BOAS1	074_ob1.wav	16	83	X
BOAS3	076_ob3.wav	66	33	X
BOAS0	081_ob0.wav	0	100	X
BOAS2	106_0b2.wav	57	42	X

Table 6.4: Classification on omitted files for the Olympus data set, recorded after exercise, 22050 frequency points.

		Classification [%]		
Class	Filename	BOAS-	BOAS+	Correct
BOAS1	029_ob1.wav	10	90	X
BOAS0	038_ob0.wav	80	20	✓
BOAS1	059_ob3.wav	0	100	✓
BOAS3	063_ob2.wav	20	80	✓
BOAS0	090_ob1.wav	15	84	X
BOAS2	103_0b0.wav	71	28	✓

7

Discussion

Here we try to analyse our results, reason why they behave as they do and point out possible errors that may have contaminated the results. We will also touch on the ethical aspects of the results and the changes it could lead to.

7.1 Hypothesis II

Hypothesis II proved to be a very good solution for the Littmann data sets as the network was able to successfully classify five out of six and four out of six of the unheard audio files respectively. We still have a limited data set which means that it is hard to say weather or not the model can classify all BOAS gradings in all dogs. But the very high classification results show that this method might be good enough to be used in a prototype application today.

The hypothesis does not work as well for the Olympus data sets. Recordings after exercise show a somewhat good result, while recordings before exercise are very poor. We do not know the reason for this. We explain a potential reason to why we get poorer results in Section 7.4.

One interesting thing to notice is that amongst all four of the models only one BOAS1 file is correctly classified worse than any of the other. We do not really know why, but it might be that it is simply harder to distinguish this class or it might point to that there are some bad files in the BOAS1 training set corrupting the training.

7.2 Hypothesis I

In the following sections we discuss the augmentation, classification and network results for our first hypothesis.

7.2.1 Augmentation

In the Chapter 5 we see that for both the Littmann and Olympus data sets, recorded before and after exercise, using the optimal settings gives the best results in most categories. Considering that these data sets use the optimal settings that have

been calculated in the Chapter 3, it is not surprising that these have the best accuracy. If we compare the Littmann and Olympus data sets, we see that a network using the Olympus data sets yield better accuracy. This is unsurprising since the Olympus records contains about 11 times more data than the Littmann. Data sets with recordings performed after exercise also show a higher accuracy than data sets with the recordings performed before exercise which is expected since the breathing sounds are more pronounced after a lighter exercise.

All data has been split into 3-seconds-segments. There is a balance here between gaining more data and the segments being long enough to contain the relevant features. We experimented with a five second segment for the Littmann data sets since the image created with 3 seconds was small. This showed promising results and is definitely a parameter that should be experimented with more in the future.

The Littmann data sets have been augmented over 900 %. The same number for the Olympus data sets is 1300 %. It would be very easy to make even larger data sets by adding more variables and values for pitch and speed shifting as well as noise to Tables 5.1 and 5.4. However, we believe that at some point we will have diminishing returns from the augmentation. A more in-depth study of the settings used during augmentation is something that would be beneficial.

7.2.2 Classification

Even though Section 5.1 show promising results we have some problems using the models to classify new recordings. Classifying BOAS2 and 3 seems to be the biggest obstacle. This can be because the amount of original data in these classes were very little. We first augment this data to have a more even number of recordings by time shifting and then continuing to augment the files with pitch and speed shifting as well as introducing noise. The data contained in these classes have been so heavily augmented to produce 483 sample segments of 24 original recordings for BOAS2 and 392 sample segments of 11 original recordings for BOAS3. Because of this, when we train our network on the augmented files, we get an overfitting problem that is not present in our accuracy and loss charts. We are unknowingly teaching our network to recognize these 24 and 11 original files, just in various forms. When we then feed a new recording of BOAS2 or 3, our model will not recognize these as BOAS2 or 3. One solution to this problem would be to have more original data. The more original data, the less augmentation is needed to have a sufficiently large data set.

As stated in Section 3.4, two ways of counteracting the overfitting problem were to limit the size of the data sets to the smallest class and using class-weights. The idea behind limiting the number of files for BOAS0, 1 and 2 to that of BOAS3 was that one class should not be more augmented than any other. The training accuracy was comparable to that of the optimal settings but the classification percentage did not improve. The attempt at using class-weights showed a preliminary decrease in both training and classification. None of these attempts provided the desired result.

7.2.3 Network

In table Table 5.12 the base-(LSTM)-network as well as network 5 and 6 performed similarly. Network 5 is larger than the base-(LSTM)-network and only performed 1 percentage point better in the maximum accuracy while the average and variance results are the same. Is this one percentage point worth the extra computations that are needed for the larger network? One could argue that the maximum accuracy is not as important as the average accuracy and that it therefore is more affordable both in time and economical factors to have as small network as possible while still achieving the highest average accuracy. Network 6 is much smaller than the base-(LSTM)-network and has the same average accuracy. This would be the preferred network if the variance was lower.

When comparing the networks for the Littmann and Olympus data sets, different networks show the best performance. The data from the different devices has different properties, for example the sample rate, which could be a factor when choosing the best network. And as we have discussed before, the amount of data gathered with the devices vary a lot and could affect which network is best suited.

7.3 Hypothesis I vs II

If we compare the classification results of the two hypotheses, Section 5.2 and chapter 6 it is clear that the second hypothesis performs better for all data sets.

If we consider the methods used, we know that we have different number of training classes. The first hypothesis uses a more complex set of four classes because of the promising results from Mårtensson [9] while hypothesis II uses only two. A network learning to distinguish between only two classes has an easier time than one having to separate four. This could be a reason for the poorer results. It would be beneficial to be able to classify all four gradings, for statistics, veterinarians and for a more precise model. But because a BOAS0 and 1 grading as well as a BOAS2 and 3 have such similar consequences, a model that can separate these two groups would also contribute to the end goal.

We also have a difference in which features we extract. When we extract the frequency feature we assume that the important information lies in the frequency data. The MFCC feature is based on frequency data but is later divided into the number of coefficients we have. Because of this we loose a lot of information even though we gain information over time when we use the LSTM network. This could also be a factor to the better performance of hypothesis II. Another advantage of using the frequency feature over MFCC is that it generates less complex training data as it is only one vector of values for each segment instead of a time series of vectors. Because of this we can use a smaller network and use less computational power.

Even though hypothesis I is much worse at classifying the omitted files it is right ever so often, but does never correctly classify a dog with BOAS1, much like hypothesis II.

7.4 Littmann vs. Olympus

The sampling rate for Littmann is 4000 Hz while Olympus has a sampling rate of 44 100 Hz. This means that the Olympus recordings have more than ten times the information that the Littmann recordings have. In hypothesis I, we counteract this somewhat by using different hop lengths, 256 and 128 respectively. This lowers the extra information to five times that of Littmann, which is still significant. This could be a reason to why the Littmann data sets have worse performance than Olympus in the first hypothesis. Different settings in segment length during the splitting augmentation can be a way to circumvent the issue but Olympus still has more information.

For our second hypothesis, the larger information is a disadvantage. Since the Olympus device has a very high sampling rate, we need to use a large array of data points. This in turn leads to many training parameters which increases the computational demands. Because of this, it was not possible to have a larger number of frequency points in this project. Increasing the frequency points could yield better results for the Olympus data sets with hypothesis II.

In Chapters 3 and 5, we see that the Littmann and Olympus data sets react differently to augmentations and networks. It is therefore questionable to have the uniform approach that this thesis has where we implement the same changes to all data sets. It could be beneficial to treat the data sets as separate projects in an effort to focus on the best solutions for the specific data set.

The Olympus recordings as well as the device resembles a mobile phone more than the Littmann does. Since one of the end goals is to implement a mobile application the Olympus data set, especially the one recorded before exercise is of large interest. Networks trained with the Olympus data sets are also the better performer of the two devices in hypothesis I. This is promising and might indicate that the Olympus data sets should be the main focus. However, Littmann will still be used by veterinarians for other ailments and is the better performer in hypothesis II, with even better classification results. There is also an issue that cell phones use different microphones making it harder to classify.

7.5 Recordings Performed Before vs. After Exercise

Recordings both before and after exercise were performed. In both our hypotheses, the Littmann data sets networks, trained with recordings before exercise performed better than the ones after exercise. This is somewhat surprising since breathing sounds should be more prominent after a lighter exercise. A possible explanation is disturbances due to the difficulty of keeping the stethoscope in the right position when the dog is agitated.

If we instead focus on the Olympus recordings. During classification with our first

hypotheses, there seems to be no difference in number of correctly classified dogs. For our second hypothesis, recordings after exercise show a much better performance than the once before.

If we study the classification results for our first hypothesis, we can see that the data sets using recordings after exercise classifies more segments as BOAS2 and 3 than the data set recorded before exercise, even though their correct label is BOAS0 or 1. This is strange and might be because our network identifies other characteristics as BOAS2 and 3 than just breathing patterns. It could for instance pick up on movements of the device since the dog is agitated.

7.6 Additional Possible Errors

The Olympus recordings have been edited, both to make them in the range of 30-60 seconds but also to remove voices and some other mayor disturbances. Multiple people have edited different files without a strict protocol. This creates a bias in the recordings that may effect how well the network may interact with new, unedited recordings. In this regard, the Littmann files are better as there exists no bias from the editors. This will hopefully be solved as more data is gathered since the same disturbances will not be present in all recordings.

When classifying a BOAS degree, veterinarians follow a strict protocol. This means that even though a recording of a dog does not show a high degree of BOAS, other factors can influence the decision. This makes it difficult for our network to classify the dog correctly. The future mobile application could also include a questionnaire which the network also takes into consideration when making its judgment.

7.7 Ethical Consideration

As there are approximately 12000 french bulldogs in Sweden alone, and if 64 % of them suffer from at least one type of BOAS affliction it would mean that at least 7500 dogs have a decreased standard of living due to a preventable disease. If we extrapolate this to other short nosed dog breeds and dogs in other countries it becomes apparent that there is a need to do something. We think that something easy to use for ordinary dog owners to assess whether they buy a healthy dog or one that will suffer throughout its life, would have a great impact on many dogs' life.

This thesis investigates a non invasive way to diagnose BOAS. If we are able to correctly classify the BOAS degree on a calm resting dog we are not exposing the dog to any additional stress. Since a future application will read recordings that could include a persons private affairs, some precautions to ensure that the data is not leaked will need to implemented in a future state.

In our a assessment the benefits of this work outweighs the risks.

8

Conclusion

We have shown that frequency data together with a CNN makes it possible to classify the BOAS grade to a high degree. Our conclusion is to use this solution for a potential product. The MFCC and other features together with a LSTM network shows that it is possible to train the network, but the limited data together with the augmentation produced a hidden over-fitting problem that showed when used on unseen files. Therefor we think that out hypothesis II is better than hypothesis I with the current amount of data. Even tho the system works with hypothesis II it could still very much benefit from more training data. To increase the accuracy and in time maybe be able to expand to more classes we recommend that recordings should continue. If limitations must be made the focus should be on the recordings before exercise.

We conclude that the Littmann and Olympus data sets behave differently to the same augmentations and networks. It could therefor be valuable to continue with two different projects in order to achieve the best results for both the Littmann and Olympus data sets.

A sufficient result without physical exercise is preferable since it offers an even easier and faster diagnosis.

8.1 Future Work

As stated before, the limited data is a large factor for the somewhat poor results. A continuation of data gathering for all four classes is crucial for the success of using machine learning to classify breathing severity in dogs.

Different variations in LSTM networks were tested as well as a CNN. A deeper study in more types of networks for the purpose of classifying breathing sounds would hopefully yield even better results than this thesis.

The work presented in this thesis will hopefully be used by both veterinarians as well as aspiring dog owners to easily get an estimate of potential breathing problems in dogs. In order to facilitate distribution, an easy-to-use mobile application would be necessary.

It is also of interest to investigate parameters such as learning rate, batch size and

number of epochs to further optimize the results. One could also further research parameters for extracting MFCC as well as class-weights.

We think that it would be beneficial to separate Littmann and Olympus into different projects because of the difference between them.

Bibliography

[1] *Genetic Welfare Problems of Companion Animals*. URL: https://www.ufaw.org.uk/dogs/german-shepherd-hip-dysplasia (visited on 06/01/2021).

[2] *Things to think about before buying a flat-faced (brachycephalic) dog*. URL: https://www.bluecross.org.uk/pet-advice/things-think-about-buying-flat-faced-dog (visited on 06/02/2021).

[3] Julia Riggs et al. "Validation of exercise testing and laryngeal auscultation for grading brachycephalic obstructive airway syndrome in pugs, French bulldogs, and English bulldogs by using whole-body barometric plethysmography". In: *Veterinary Surgery* 48 (Jan. 2019). DOI: 10.1111/vsu.13159.

[4] *Statistik ur hundregistret - Jordbruksverket.se*. May 2021. URL: https://jordbruksverket.se/e-tjanster-databaser-och-appar/e-tjanster-och-databaser-djur/hundregistret/statistik-ur-hundregistret.

[5] Alice Nordevik. "Här är valpboomens mest populära raser". In: (May 2021). URL: https://www.svt.se/nyheter/har-ar-valpboomens-popularaste-raser (visited on 05/18/2021).

[6] Ida Bertilsson and Linda Keeling. *Phenotypic variation for BOAS within four brachycephalic dog breeds-Can good welfare be obtained? Fenotypisk variation for BOAS within four brakycefala hundraser-Kan god djurvälfärd uppnås?* 2019. URL: https://stud.epsilon.slu.se.

[7] Brian McFee et al. "librosa/librosa: 0.8.0". In: (July 2020). DOI: 10.5281/ZENODO.3955228. URL: https://doi.org/10.5281/zenodo.3955228#.YJFZeQt24SM.mendeley.

[8] Keras. *Keras layers API*. Available at https://www.thermofisher.com/order/catalog/product/R37601#/R37601[Accessed 2021-03-05].

[9] Mårtensson Moa.

[10] *3M™ Littmann® Electronic Stethoscope Model 3200*. URL: https://www.littmann.com/3M/en_US/littmann-stethoscopes/products/~/3M-Littmann-Electronic-Stethoscope-Model-3200/?N=5932256+8711017+3293188392&rt=rud (visited on 05/07/2021).

[11] *Linear PCM Recorder LS-P1*. URL: https://asia.olympus-imaging.com/product/audio/lsp1/spec.html (visited on 05/07/2021).

[12] *Overview — NumPy v1.20 Manual*. URL: https://numpy.org/doc/stable/ (visited on 05/07/2021).

[13] *librosa.effects.pitch_shift — librosa 0.8.0 documentation*. URL: https://librosa.org/doc/main/generated/librosa.effects.pitch_shift.html (visited on 05/07/2021).

[14] *librosa.effects.time_stretch — librosa 0.8.0 documentation*. URL: https://librosa.org/doc/main/generated/librosa.effects.time_stretch.html (visited on 05/07/2021).

[15] Jason Brownledd. *SMOTE for Imbalanced Classification with Python*. Jan. 2021. URL: https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/.

[16] *The dummy's guide to MFCC*. URL: https://medium.com/prathena/the-dummys-guide-to-mfcc-aceab2450fd (visited on 05/07/2021).

[17] A.V. Oppenheim and Ronald Schafer. "From Frequency to Quefrency: A History of the Cepstrum". In: *Signal Processing Magazine, IEEE* 21 (Oct. 2004), pp. 95–106. DOI: 10.1109/MSP.2004.1328092.

[18] *librosa.feature.mfcc — librosa 0.8.0 documentation*. URL: https://librosa.org/doc/latest/generated/librosa.feature.mfcc.html (visited on 05/07/2021).

[19] *librosa.feature.rms — librosa 0.8.0 documentation*. URL: https://librosa.org/doc/latest/generated/librosa.feature.rms.html#librosa.feature.rms (visited on 05/07/2021).

[20] Costas Panagiotakis and Georgios Tziritas. "A speech/music discriminator based on RMS and zero-crossings". In: *Multimedia, IEEE Transactions on* 7 (Mar. 2005), pp. 155–166. DOI: 10.1109/TMM.2004.840604.

[21] *librosa.feature.zero_crossing_rate — librosa 0.8.0 documentation*. URL: https://librosa.org/doc/latest/generated/librosa.feature.zero_crossing_rate.html (visited on 05/07/2021).

A

BOAS Classification Protocol

Date: 2020-09- _____ Location: _____

Dog namn:		**Breed:** **male/female**
Name of owner:		**Neutered: Yes No**
E-mail adress:		**Microchip number:**
Date of birth:	**Weight: _____kg** **Body condition score (BCS): _____/9**	**Colour:**

Medical History: No Yes :

Medications: : No Yes: .....

Do you perceive that your dog has abnormal breathing sounds:

Never Seldom (once a month) Often (several times a week) Daily but intermittent

Constant Only when asleep

Does your dog have trouble breathing:

Never Seldom (once a month) Often (several times a week) Daily but intermittent

Constant Only when asleep Only when it is warm outside

Does your dog usually sleep on:

lying on the back lying on the side lying on the chest/belly in a sitting position

Does your dog sleep with:

a normal head position an elevated head position a toy in its mouth

Does your dog have episodes of apnea (periodically not breathing at all/holding its breath) during sleep?

No Yes

If Yes, how often?

Seldom (happened once or twice) Rare (happens monthly) Often (happens weekly) Daily

Does your dog ever had episodes of collapse?

No Yes

If Yes, how often?

Seldom (happened once or twice) Rare (happens monthly) Often (happens weekly)

If Yes, when does it happen?

during rest during exercise both during rest and exercise

Does your dog wake up/disturb frequently during night/sleeping cycles? No Yes

Other information:

☐ Date: 2020-09- _____ Location: _____
Test id: _____ BOAS grading set by:_____
Photo dog ☐ ☐ /nostrils By:_____
Film respiratory pattern By:_____
Recordings before ET _____ Panting: _____Operator:_____
☐ Recordings after ET_____ Panting: _____Operator: _____

Physical examination Pre ET

- **Stress level:** Normal Mild Moderate Severe
- **Open mouth breathing: :** No Intermittent Constant
- **Nostrils:** Open Mild stenosis Moderate stenosis Severe stenosis
- **Stertors (low pitch noise):** Not audible Mild Moderate Severe
- **Stridors (high pitch noise):** Not audible Mild Moderate Severe
- **Inspiratory effort:** Not present Mild Moderate Severe
- **Expiratory effort:** Not present Mild Moderate Severe
- **Cyanosis and/or syncope: :** No Yes _____
- **Heart auscultation:** Normal Abnormal _____
- **Lung auscultation:** Normal Abnormal _____
- **Cough reflex on tracheal palpation:** No Yes Cough time:_____seconds
- **Nose level** below lower eyelid level with lower eyelid above lower eyelid
- **Entropion** No Yes
- **Epiphora** No Yes
- **Cornea ulcus** No Yes

Physical examination Post ET

- **Open mouth breathing: :** No Intermittent Constant
- **Nostrils:** Open Mild stenosis Moderate stenosis Severe stenosis
- **Stertors (low pitch noise):** Not audible Mild Moderate Severe
- **Stridors (high pitch noise):** Not audible Mild Moderate Severe
- **Inspiratory effort:** Not present Mild Moderate Severe
- **Expiratory effort:** Not present Mild Moderate Severe
- **Cyanosis and/or syncope: :** No Yes _____
- **Heart auscultation:** Normal Abnormal _____
- **Lung auscultation:** Normal Abnormal _____

BOAS Functional Grading: Grade 0 Grade I Grade II Grade III

B

Networks

B.1 Network 2

Table B.1: A network based on the base-(LSTM)-network but with fewer neurons in the LSTM layers.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 24, 512)	input dependent
dropout (Dropout=0.3)	(None, 24, 512)	0
lstm_1 (LSTM)	(None, 126)	322056
dropout_1 (Dropout)	(None, 126)	0
dense (Dense)	(None, 32)	4064
dropout_2 (Dropout=0.3)	(None, 32)	0
dense_1 (Dense)	(None, 4)	132

B.2 Network 3

Table B.2: A network based on the base-(LSTM)-network but with an extra LSTM network and fewer neurons.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 24, 512)	input dependent
dropout (Dropout=0.3)	(None, 24, 512)	0
lstm_1 (LSTM)	(None, 24, 256)	787456
dropout_1 (Dropout=0.3)	(None, 24, 256)	0
lstm_2 (LSTM)	(None, 128)	197120
dropout_2 (Dropout)	(None, 128)	0
dense (Dense)	(None, 32)	4128
dropout_3 (Dropout=0.3)	(None, 32)	0
dense_1 (Dense)	(None, 4)	132

B.3 Network 4

Table B.3: A network based on the base-(LSTM)-network but with higher dropout rate.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 24, 1024)	input dependent
dropout (Dropout=0.5)	(None, 24, 1024)	0
lstm_1 (LSTM)	(None, 256)	1311744
dropout_1 (Dropout=0.5)	(None, 256)	0
dense (Dense)	(None, 32)	8224
dropout_2 (Dropout=0.5)	(None, 32)	0
dense_1 (Dense)	(None, 4)	132

B.4 Network 5

Table B.4: A network based on the base-(LSTM)-network but with an extra LSTM layer.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 24, 1024)	input dependent
dropout (Dropout=0.3)	(None, 24, 1024)	0
lstm_1 (LSTM)	(None, 24, 256)	1311744
dropout_1 (Dropout=0.3)	(None, 24, 256)	0
lstm_2 (LSTM)	(None, 256)	525312
dropout_2 (Dropout=0.3)	(None, 256)	0
dense (Dense)	(None, 32)	8224
dropout_3 (Dropout=0.3)	(None, 32)	0
dense_1 (Dense)	(None, 4)	132

B.5 Network 6

Table B.5: A smaller network; one LTSM layer which is half as big.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 47, 512)	input dependent
dropout (Dropout=0.3)	(None, 47, 512)	0
lstm (LSTM)	(None, 47, 512)	4202496
dropout (Dropout=0.3)	(None, 47, 512)	0
dense (Dense)	(None, 32)	8224
dropout_2 (Dropout=0.3)	(None, 32)	0
dense_1 (Dense)	(None, 4)	132

B.6 Network 7

Table B.6: Much fewer nodes per level

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 47, 256)	input dependent
dropout (Dropout=0.3)	(None, 47, 256)	0
lstm (LSTM)	(None, 64)	82176
dropout (Dropout=0.3)	(None, 64)	0
dense (Dense)	(None, 16)	1040
dropout_2 (Dropout=0.3)	(None, 16)	0
dense_1 (Dense)	(None, 4)	68

B.7 freq2

Table B.7: A bigger CNN to work with basic frequency data.

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 1, 2048)	input dependent
conv1d (Conv1D)	(None, 1, 1024)	14681088
dropout (Dropout=0.3)	(None, 1, 1024)	0
conv1d (Conv1D)	(None, 1, 512)	2621952
dropout (Dropout=0.3)	(None, 1, 512)	0
conv1d (Conv1D)	(None, 1, 128)	327808
dense_1 (Dense)	(None, 4)	516

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY