# Exploring Deep Learning Approaches to Cleft Lip and Palate Speech

Can neural networks be used to accurately evaluate velopharyngeal competence?

Tofig Mamedov, Joel Bluhme

Master's thesis
2021:E69

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematical Statistics

# Abstract

Cleft lip and palate belong to the most common deformities present at birth. The condition hampers normal speech development in children, and treatment involves both surgery and regular sessions with a speech pathologist. The speech pathologist assesses the child's speech impairment stemming from the condition on a three-point scale: "*Competent*", "*Marginally incompetent*" and "*Incompetent*" and the rating forms a basis for future treatment decisions. This procedure is time and resource intensive since close examination of the entire recording is necessary for an accurate aggregate rating. Furthermore, field experience is that the assigned rating for a singular recording can be biased and the ratings from different speech pathologists are inconsistent.

In this thesis, deep learning methods are used to classify audio recordings of children into the three categories. The ambition of this undertaking is to rid the classification of bias and provide speech pathologists with a consistent baseline rating. Different steps in the pre-processing of speech therapy recordings are explored to transform the raw audio input into meaningful information for a neural network. The best performing network structure was a convolutional neural network model and it manages to classify recordings with a 89.76% accuracy by using Mel-spectrograms on 0.2 seconds of pre-processed audio segments. Recommendations about further work is discussed with the end goal of developing a fully automatic classifier with appropriate data gathering methods.

**Keywords**: Cleft Lip, Cleft Palate, Speech, CNN, RNN, LSTM, Deep Learning

# Sammanfattning

Läpp-, käk- och gomspalt tillhör de vanligaste missbildningarna vid födseln. Tillståndet hämmar normal talutveckling hos barn, och behandling innebär oftast en eller flera operationer samt regelbundna sessioner med en logoped. Logopeden bedömer barnets talförmåga på en trepunktsskala: "*Kompetent*", "*Marginellt inkompetent*" och "*Inkompetent*" och betygsättningen utgör en grund för framtida behandling. Denna procedur är tids- och resurskrävande eftersom noggrann granskning av hela inspelningen är nödvändig för en korrekt sammanlagd bedömning. Vidare visar praktisk erfarenhet att bedömningen av en inspelning kan vara partisk och betyg från olika logopeder upplevs delvis som inkonsekventa.

I denna avhandling används djupinlärningsmetoder för att klassificera ljudinspelningar av barn i de tre kategorierna. Ambitionen med detta är att ge en objektiv bedömning som logopeder kan använda som en stödjande basvärdering. Olika steg i förbehandlingen av inspelningarna från talterapi-samtalen undersöks för att omvandla inspelningarna till meningsfull information för ett neuralt nätverk. Den bäst presterande nätverksstrukturen var en CNN som lyckades klassificera inspelningar med en noggrannhet på 89.76% med hjälp av Mel-spektrogram på 0.2 sekunder långa ljudsegment. Rekommendationer kring vidare arbete diskuteras med slutmålet att utveckla en helautomatisk lösning.

**Nyckelord**: Läppspalt, käkspalt, gomspalt, Tal, CNN, RNN, LSTM, Djupinlärning

# Acknowledgements

We would like to express our sincere gratitude to Magnus Becker, Måns Cornefjord and Susanna Whitling for providing data and helping us navigating the intricacies of cleft lip and palate treatment procedures. This project would however never be possible without the weekly guidance of our supervisor Andreas Jakobsson. His interest in our progress, commitment to the project as a whole and frequent availability made this effort reach a satisfying conclusion.

Joel Bluhme, Tofig Mamedov, Lund, June 2021

# Contents

# List of Figures

# List of Tables

# 1
# Introduction

In this chapter, the motivating context of this thesis is presented. The classification procedure of cleft lip and palate speech is described, and the difficulties of making an accurate rating are discussed. Finally, the goals and ambitions of this project are specified.

## 1.1 Context and Motivation

Cleft lip and palate (CLP) are two of the most common deformities present in children at birth. Every year, about 150-200 children are born with CLP in Sweden, statistically representing 1 in 500 births. CLP is discovered during pregnancy when it is detected by ultrasound. It is important to note that CLP is an umbrella term describing several different conditions. For example, there is a distinction between *complete* cleft palate, meaning that both the hard and soft palate (roof of the mouth) are not completely joined, and *incomplete* cleft palate meaning that only the soft palate is cleft. Another distinction is between unilateral and bilateral cleft lip, depending on if only one or both sides of the lip are split [13]. In Figure 1.1, three different conditions of CLP are presented. Children with CLP undergo one or several surgeries for cosmetic reasons and alleviate difficulties with speaking and eating. Even after surgical interventions, discrepancies can still exist if compared to normal speech development for children. This is due to velopharyngeal insufficiency, which in simple terms means that the velum fails to close properly against the pharyngeal wall located in the back of the throat. Excluding intended nasal sounds (e.g.,*/m/, /n/*), proper closure is vital in human speech production since it is required to force air from the vocal tract to escape through the oral cavity instead of the nasal cavity. Consequently, the palate remaining partially open for children with CLP commonly produces hyper-nasal speech due to speech escaping through the nose. For that reason, children with CLP regularly see a speech pathologist from their first year. The hope is to assess and remedy speech errors as much as possible, intending to complete treatment before the child is set to start school.

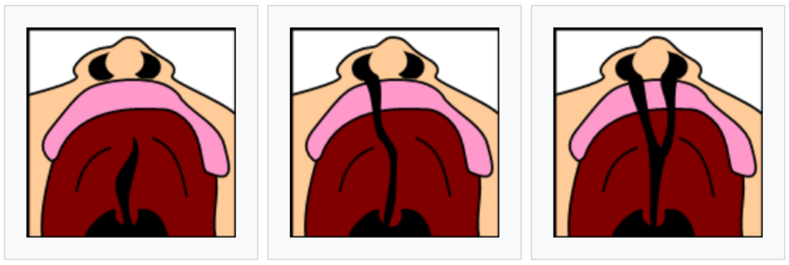

**Figure 1.1:** From left to right: Incomplete cleft palate, Unilateral complete lip and palate, Bilateral complete lip and palate [54] [52] [53].

In the speech therapy sessions, the child is asked to repeat words and phrases constructed to give away the severity of speech impairment. As an aid in this pursuit, several tests and routines consisting of such words and phrases have been written in several languages. In Sweden, Svenskt Artikulations- och Nasalitetstest (SVANTE) is commonly used by speech pathologists to test the child's speech competence regarding, e.g., articulation, phonology, and nasality. Since hearing and assessing articulations in real-time is a rather challenging task, the sessions are recorded later assessed more thoroughly by the speech pathologist. According to the assessment working sheet for SVANTE, a number of rules apply when a speech pathologist is listening to a recording [13]:

1. The evaluation session is expected to take around 45 minutes per recording
2. Do not listen if you are tired
3. Listen to each articulation as many times you find necessary
4. Transcribe everything you assess

The aggregated rating of the entire recording is *Velopharyngeal competence*, and it aims to describe the severity of speech impairment stemming from CLP. It is rated on a scale from 1 to 3, where 1 indicates little to no impairment while 3 indicates severe impairment. This score forms a basis for future treatment (speech training and, possibly, further surgery) suits best for the individual in question. As can be imagined, listening to and classifying the recordings are time-consuming tasks for the speech pathologist. Moreover, each recording is cross-validated by a number of speech pathologists to improve the likelihood of correct classification. The rating scale was initially constructed to have 5 levels. Still, in practice, it was noted that the opinion from one speech pathologist could differ to an undesirable degree when assessing the same recording, and hence the scale was made coarser. Similar issues do, despite the simplified scale, still exist. In some instances, it has been noted that the same speech pathologist assigns different ratings to the same recording, re-listening to the recording at a later date. Since the classification carries great importance in the future care plan for the child, it is vital that it is performed as accurately as possible.

## 1.2 Project Goals and Ambitions

Because of the time and resource-consuming nature of classifying CLP audio recordings, there is a strong incentive to automate the listening and rating processes. Furthermore, as previously discussed, undesired degrees of subjectivity and variance exists in the assessment made by the speech pathologists. For that reason, it would be of interest to develop a classification algorithm that takes as input a recording of a speech therapy session and outputs a baseline estimate of the velopharyngeal competence. If this algorithm is based on cross-validated ratings from multiple speech pathologists, it could, if effectively constructed, help speech pathologists in their day-to-day work. By extension, it gives medical professionals a better understanding of what (if any) further surgical interventions are necessary. Furthermore, an algorithm is not bound by the physical capabilities of the human ear and mind when classifying speech data. Therefore, it can capture aspects of the voice that a speech pathologist cannot and provide an instantaneous evaluation.

In this project, we aim to explore different deep learning approaches to develop an accurate classification algorithm. Deep learning has been shown to excel in complex tasks when dealing with inputs consisting of big, unstructured, and complex data. Successful implementations exist within many fields, such as visual and image recognition, speech recognition, and written language processing [39]. Training and implementing a deep learning network is not a straightforward task. It requires a number of different steps to be taken, including pre-processing, feature extraction, network architecture design, and training. Our aim with this project is to explore different paths and approaches concerning these steps in the context of classifying CLP speech. Moreover, this thesis is the first step in what is, hopefully, a long and prosperous project. Therefore we will strive to document our work as much as possible to make the lives of the next team much easier. The purposes of this thesis can be summarized as:

1. Build an automatic CLP speech algorithm based on deep learning approaches with as high accuracy as possible
2. Provide recommendations with regards to data collection, pre-processing and network structure, etc.
3. Leave a solid working toolkit and documentation for future developers working on the algorithm

## 1.3 Resources and Software

The majority of code for this project was written in `MATLAB R2021a`, installed with the Deep Learning and Audio Toolbox. A portion is also written in `Python`, using the `PyTorch` library for deep learning applications. Since some aspects of the code require long intensive calculations, more than can be handled on a regular desktop PC, access to the Alvis computing cluster and its GPUs was also granted for this project. Alvis is a GPU cluster based in Gothenburg, made for running computationally intensive job scripts necessary in deep learning applications [46].

# 2
# Voice and Speech

As the fundamental communicative tool for humans, the voice plays an integral role in everyday life. The human vocal production system can produce a remarkably wide array of phonemes that, when combined, form words and sentences. As such, our voice constitutes explicit commutation in the form of speech content. Moreover, in addition to direct speech, the voice often conveys implicit information about the speaker, including biological information (e.g., age and gender) and paralinguistic information (e.g., emotions and social status) [21]. In this chapter, we briefly cover an overview of the anatomy behind human vocal production, outline how voice and speech can be represented in mathematical notation, and present a model commonly used to describe the generation of speech output. Lastly, some frequently observed speech characteristics for children with cleft lip and palate are discussed.

## 2.1 The Vocal Production System

To better understand how different sounds are produced, it is helpful to describe the human vocal production system's anatomy briefly. A diagram of the system can be seen in 2.1.

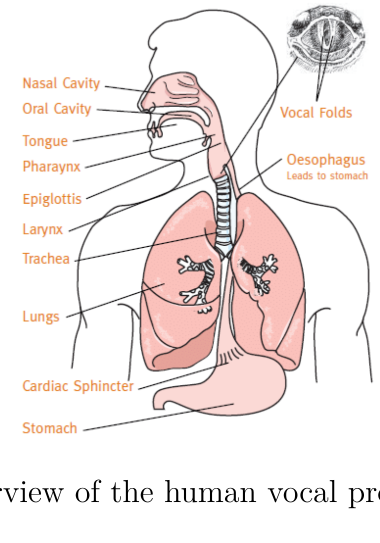

**Figure 2.1:** Overview of the human vocal production system [17].

The system can roughly be divided into three sections: the lungs and lower airways, the vocal folds, and the articulators. The lungs produce airflow and air pressure, which is transported through the lower airways to the vocal folds. When the airflow reaches the vocal folds in the larynx, it causes them to vibrate, generating air pressure fluctuations - sound. The vocal folds are elastic, and their length and tension can be controlled by surrounding musculature, which affects the characteristics of their vibrations. For example, by building up a higher vocal fold tension, the pitch or tone of the produced sound becomes higher. After passing through the vocal folds, the resulting airflow is further modified and shaped by the articulators above the larynx (e.g., lips, teeth, palate, tongue, and velum). The way the articulators modify the sound waves is determined through the resonances in the vocal tract, which are altered by changing the shape and positions of the articulators. For example, by pressing the lips tight together while using the tongue to restrict the airflow to a specific range, humans can create a whistling sound [36].

There are several ways one can classify and describe the resulting speech output from the system described above. A broad and commonly used distinction is between voiced and unvoiced speech. The former consists of phonemes produced through vocal fold vibration, for example vowels such as */a/, /e/, /u/, /i/* and voiced consonants such as */b/, /d/, /g/.* Unvoiced speech, on the other hand, refers to phonemes produced predominantly without vibration of the vocal folds and includes the stop consonants, for example, */p/, /t/, /k/* [19]. In this case, the sound is generated instead through turbulence at one or more air passages in the mouth cavity, for example by pressing the tip of the tongue against the palate and the inside of the upper teeth and producing a */t/.* The dichotomy between voiced and unvoiced speech is, however, not a strict one since there exists a number of sounds which fall under both categories. Consider for example the */v/* in $\underline{\text{v}}$ision, which is both voiced and unvoiced since it is generated through vocal fold vibration while at the same time being a result of turbulence as air passes through the tightly pressed together upper teeth and the lower lip.

In the scientific domain of speech-language pathology, virtually every sound that the human vocal system is capable of producing belongs to one or more classes. Consider, for example, the fricatives which are consonants produced when the airflow is forced through two articulators pressed tight together or nasals which are consonants constructed by letting the airflow pass through the nose instead of the mouth [45]. Moreover, each class has several sub-classes, which are often based on how the articulators generate the sound. As an example, consider the nasal class, which has 3 sub-classes or different phonemes pertaining to which articulators are blocking the airflow to escape through the mouth and forcing it through the nasal cavity:

- **Bilabial nasals**: Both lips, e.g. */m/* in $\underline{\text{m}}$ode
- **Alveolar nasals**: Tongue and alveolar ridge, e.g. */n/* in $\underline{\text{n}}$eck
- **Velar nasals**: Tongue and velum (soft palate), e.g. */ŋ/* in so$\underline{\text{ng}}$

With this basic understanding of how human vocal production works, we can start approaching common ways to model the output of this complex and multifaceted system.

## 2.2 Time and Spectral Representations

When speech emerges from the speaker's mouth, it gives rise to fluctuations in air pressure. This raw signal is analog since it is a continuous and time-varying signal. Analog audio recordings, such as vinyl and cassette tapes, have many desirable features compared to digital recordings. For example, it is able to depict the source audio more accurately. However, today most audio is recorded and stored in digital form, which has a number of advantages. Perhaps most notably, storing audio as a digital signal allows for an easier and higher degree of compression and thus more efficient storing of the audio [28]. Digital recording devices are systems which can transform and store the information contained in analog sound signal in a digital format. To store an analog signal digitally, it needs to go through two main processes: time discretization (sampling) and amplitude discretization (quantizing). The former maps the time-continuous analog signal onto an equally spaced discrete-time grid by sampling it at a predetermined sampling rate $f_s$. The amplitudes of these samples are continuous, but through the amplitude discretization, they are approximated to a finite number of levels. The analog to digital signal conversion process is outlined in Figure 2.2.

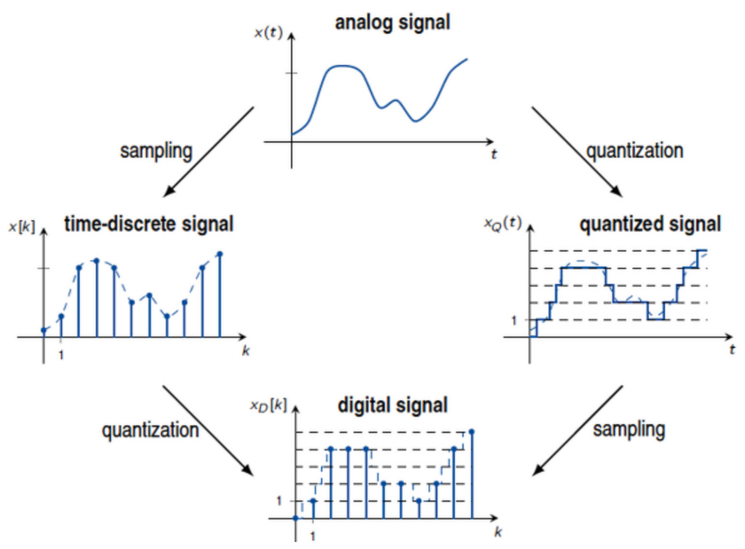

**Figure 2.2:** Transforming an analog signal into a digital signal [47].

### 2.2.1 Time-domain Representation

The produced digital sound through the conversion steps described above can be represented in the time-domain as a real-valued vector or sequence $\boldsymbol{x}[n], n = 0, \ldots, N-1$ corresponding to the measurement of air pressure of the specific sound recorded at the sampling rate $f_s$ measurements per seconds. If the recording was made in stereo, the recording consists of two vectors of equal length $\boldsymbol{x}_1[n]$ and $\boldsymbol{x}_2[n]$ corresponding to each channel. In the time domain, audio signals are most commonly represented as a waveform in which the amplitude is plotted against time. In Figure 2.3, the waveforms for an (adult) female speech pathologist and a 5-year-old with cleft palate uttering the phrase "*Kicki kokar korv*" are plotted. This phrase is common in speech therapy sessions, and it aims to test the child's ability to produce the voiced consonant */k/*.

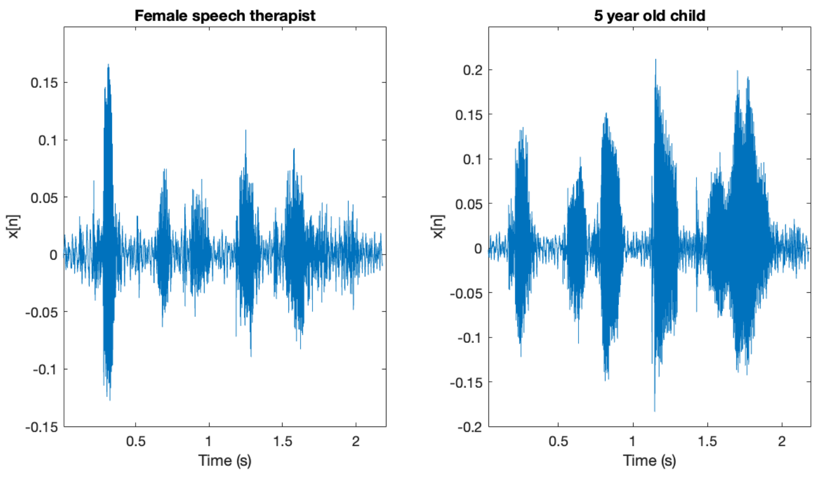

**Figure 2.3:** Waveforms for the phrase "*Kicki kokar korv.*"

Upon visual inspection of the plots in Figure 2.3 one can tell that the general outline of both waveforms is similar, which intuitively makes sense. Both individuals spoke the same words and one, therefore, expects energy in around similar positions time-wise. The amplitude is higher for the child in absolute terms compared to the speech pathologist. When playing back the recording, this is reflected as a higher volume for the child compared to the therapist (the microphone was placed closer to the child since he/she is the patient). There exists a number of features in the time-domain commonly used to characterize the audio signal. Some of the most applied ones are defined and discussed briefly below.

**Energy** Let $x_i(n), n = 1, \ldots, W_L$ be the $i$th sub-sequence or *frame* taken from the audio vector $\boldsymbol{x}[n]$. Then the energy of this sub-sequence is defined as

$$E(i) = \sum_{n=1}^{W_L} |x_i(n)|^2 \tag{2.1}$$

It is common to divide (2.1) by the length of the sub-sequence $W_L$, thus normalizing the energy and removing dependency on frame length. Doing so yields the *power* of the frame. In the context of speech classification and recognition, energy and power often serve as a primary, although somewhat naive, indicators if speech is present in the signal. This is because if $\boldsymbol{x}[n]$ is a speech signal, it is expected to exhibit high fluctuation in its short-term energy envelope due to varying intensity in speech signals and short periods of silence between phonemes.

**Zero-Crossing Rate** Let again $x_i(n), n = 1, \ldots, W_L$ be the $i$th sub-sequence or *frame* taken from the audio vector $\boldsymbol{x}[n]$. Then the Zero-Crossing Rate (ZCR) is defined as

$$Z(i) = \frac{1}{2W_L} \sum_{n=1}^{W_L} \left| \operatorname{sgn}\left[x_i(n)\right] - \operatorname{sgn}\left[x_i(n-1)\right] \right| \tag{2.2}$$

in which $\operatorname{sgn}(\cdot)$ denotes the sign function. By dividing the number of times the signal changes from negative to positive (and vice versa) in the $i$th frame of the audio signal with the length of the frame, ZCR serves as an indication of the noisiness of that particular frame. As such, it can be seen as a rough indication of certain spectral characteristics, which will be covered below.

### 2.2.2 Frequency-domain representation

Alongside the time-domain representation and corresponding features, one often turns to the frequency domain to analyze an audio signal. Just like the time-domain representation $\boldsymbol{x}[n]$, the frequency domain examines the amplitude of the signal. However, it does so as a function of frequency instead of time. To derive a frequency-domain (a spectral) representation of the audio signal $\boldsymbol{x}[n]$, a *Discrete Fourier Transform* (DFT) is applied to the signal. The DFT has been called the backbone of modern signal processing, and its applications play a vital role in almost all signal processing systems since it is the only Fourier representation that computers can evaluate [22]. The DFT of a discrete-time signal, for example, the $N$ samples long digital audio signal $\boldsymbol{x}[n]$ is given by

$$X(k) = \sum_{n=0}^{N-1} \boldsymbol{x}[n] e^{-i\frac{2\pi}{N}kn} = \sum_{n=0}^{N-1} \boldsymbol{x}[n] \cdot \left[\cos\left(\frac{2\pi}{N}kn\right) - i \cdot \sin\left(\frac{2\pi}{N}kn\right)\right] \tag{2.3}$$

where $k = 0, \ldots, N-1$ and $i$ is the imaginary unit. In (2.3) we can see two different but equally accurate representations of the spectral contents of the signal $\boldsymbol{x}[n]$. The first one indicates that the output of the applied transform is a sum of $N$ complex-valued coefficients. The second representation hints at the very core point of spectral analysis of an audio signal, namely decomposing a signal into its sinusoidal components and revealing information about these harmonics' amplitude and phase. Applying the *inverse* DFT to the complex-valued coefficients $X(k)$ returns an exact reconstruction of the original audio signal:

$$\boldsymbol{x}[n] = \frac{1}{N} \sum_{n=0}^{N-1} X(k) e^{i\frac{2\pi}{N}kn}, \quad n = 0, \ldots, N-1 \tag{2.4}$$

As such, we see how both the time-domain discrete signal $\boldsymbol{x}[n]$ and the complex-valued $X(k)$ are equally valid representations of the underlying signal but in two different domains. A practical interpretation of the DFT coefficients $X(k)$ is that given the sample rate $f_s$, which was used to obtain the discrete-time representation $\boldsymbol{x}[n]$ of the analog signal, then the $k$th exponential corresponds to the frequency $f_k = k\frac{f_s}{N}$. Hence, naturally, a larger $N$ (a longer signal $\boldsymbol{x}[n]$) should, theoretically, result in a more dense and accurate representation of the frequency domain since $\frac{f_s}{N}$ gets smaller as $N$ increases, thus increasing frequency *resolution*. However, this is only valid as long as the distribution of the process which generated the observed signal remains *stationary*. Ignoring, for now, the mathematical formalism of the stationary feature, consider a simple example of a 3-second long audio recording consisting of a short conversation (2 s long) followed by a music segment (1 s long). It is clear that, e.g., the intensity of the signal is not consistent throughout this recording. In a simplified way, this indicates a change of stationarity since the properties of the signal have shifted from one state to another. Even recordings consisting exclusively of speech cannot be considered stationary over extended periods of time. For the DFT in (2.3) to accurately depict the signal's spectral content, it must hold that the signal (or, more formally, the process generating the signal) is stationary. In voice and speech analysis, it is often assumed that while the signal itself is non-stationary, it can be approximated as *wide-sense stationary* (WSS) if analyzed in short time frames of 20-30 ms. A random process $X(t)$ is WSS if its mean and autocorrelation function (ACF) is time-invariant and if its variance is finite [18]:

1. $E\left[X\left(t\right)\right] = \mu_X$, the mean does not depending on $t$
2. $r_X(t_1, t_2) = r_X(t_2 - t_1) = r_X(\tau)$, the ACF only depends on the time-step
3. $V\left[X(t)\right] = E\left[X\left(t\right)^2\right] - \mu_X^2 = \sigma_X^2 < \infty$

To maintain that at least WSS holds, it is common to compute the DFT in shorter time frames better to capture the local frequency content present in the signal. One can then study the sequence of generated spectral contents to understand how the frequencies of the signal changes over time. The *Short-Time Fourier Transform* (STFT) does exactly this for the signal $\boldsymbol{x}[n]$:

$$X(k, m) = \sum_{n=0}^{N-1} \boldsymbol{x}[n] w[n-m] e^{-i\frac{2\pi}{N}kn} \quad , k, m = 0, \ldots, N-1 \tag{2.5}$$

where $w[n]$ is a fixed-size and time-shifted *windowing function* of appropriate choice. As such, $X(k_i, m_j)$ denotes the $i$th Fourier coefficient for the $j$th frame of the entire signal $\boldsymbol{x}[n]$. The main purpose of the multiplication with the windowing function $w[n]$ is to ensure that only contributions for the $m$th frame are analyzed. The simplest choice of windowing function, the rectangular window, achieves this by taking the value 1 for the values corresponding to the $m$th frame and 0 elsewhere. In practice, especially in speech analysis, one often relies on a more sophisticated choice of windowing function, for example the Hanning window which is constructed as $w[n] = 0.5\left[1 - \cos\left(\frac{2\pi n}{N}\right)\right] = \sin^2\left(\frac{\pi n}{N}\right)$. The sampling of the original signal in the time domain results in unwanted distortions in the estimated spectrum known as leakage when applying the rectangular window due to its relatively high sidelobes. The Hanning window has a wider main lobe but faster-decreasing sidelobes, which prevent spectral leakage, i.e., inaccurate depictions of the true spectrum [51]. The *spectrogram* is a visual representation of the spectral contents of a signal as it varies over time. It is calculated simply by squaring the STFT of an audio signal and is most commonly plotted as an image where the frequency intensity variations are indicated by changing colors. Below in Figure 2.4, the spectrograms for the adult and child uttering the phrase "*Kicki kokar korv*" are plotted.

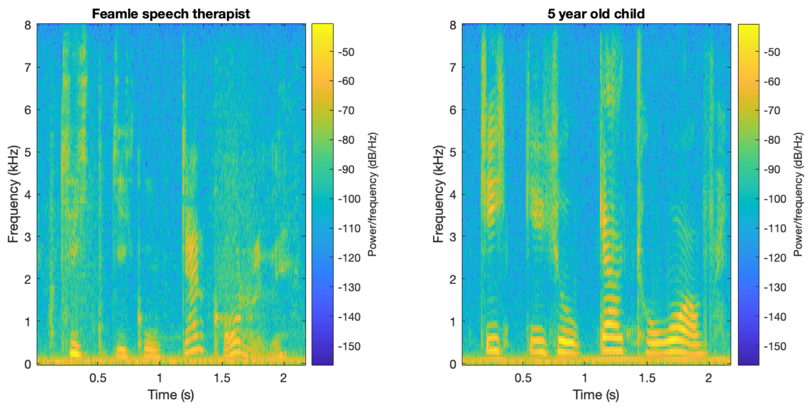

**Figure 2.4:** Spectrograms for the phrase "*Kicki kokar korv*".

Inspecting the two plots closely, it is clear that just like the waveforms in Figure 2.3 the two spectrograms resemble each other structurally. One can, however, note that the frequency content for the child is, on average, higher than for the adult. This is due to the fact that the child's vocal folds are not fully grown and thus shorter than the adult's, therefore producing airwaves with lower periodicity (higher frequency) when vibrating. As previously established, the time-domain and frequency-domain representations are equally valid representations of the underlying signal. In Figure 2.5, the two domains are plotted in the same graph along one axis each to visualize how the spectral content varies with time.

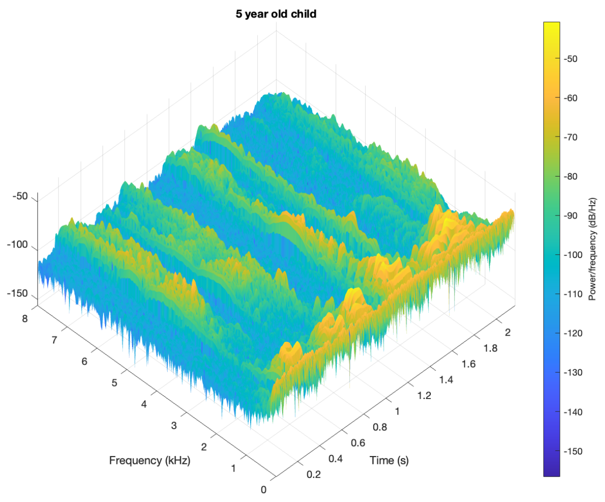

**Figure 2.5:** Time and spectral representation for the phrase "*Kicki kokar korv*".

Generally speaking, features extracted from the frequency domain can capture more complex and varied characteristics in the underlying audio content when compared to features extracted in the time domain [10]. This has spurred the development of handcrafted spectral features commonly used to characterize and discriminate the content in the audio file. Below follows a brief presentation of the most prominent ones within the speech analysis field.

**Pitch** Also known as the *fundamental frequency* $f_0$, the pitch is the frequency that the vocal folds vibrate at. As discussed in section 2.1, humans can adjust speech pitch through the tension in the vocal folds. The range of human pitch is around 85 to 225 Hz, where men usually exhibit pitches in the lower domain of that range and women usually in the higher domain [36]. Several different algorithms have been developed to estimate pitch, for example, the YIN algorithm and the MPM algorithm, which both use estimates of the autocorrelation function of the speech signal to generate a pitch estimation [5], [26].

**Harmonics** The fundamental frequency $f_0$ gives rise to harmonics, or overtones, which are integer multiples of the fundamental frequency. They are the result fundamental frequency resonances. Similar to pitch, the harmonics are affected by anatomical configuration - generally a long and wide vocal tract results in enhanced lower harmonics and hence a darker and fuller voice while a shallower vocal tract results in higher resonances being more prominent, generating a brighter voice output [42]. This can be seen when comparing the spectrograms in Figure 2.4. The child clearly has more prominent resonances at the higher frequency bands when compared to the adult.

**Formants** A closely related concept to harmonics are formants. While a harmonic structure of overtones is usually observed in the spectrum of any audio signal, formants are more related to speech analysis. More specifically, the formants are the vocal tract and articulator specific resonance frequencies. Because of the abnormal resonances in the vocal tract caused by CLP, the formants can have different locations in the frequency domain for CLP-children. A common approach to formant estimation is Linear Predictive Coding (LPC).

**VLHR** Voice low tone to high tone ratio (VLHR) is a feature that has been used and proven to be statistically significant as a measure for hypernasality. For an audio signal it is computed as log-transformed ratio:

$$\mathrm{VLHR} = 10\log_{10}\left(\frac{\mathrm{HFP}}{\mathrm{LFP}}\right) \tag{2.6}$$

where HFP is the power in the frequency band above a pre-determined threshold and LFP is the power in the band below the threshold. Common threshold choices range from 400-600 Hz, depending on application and language. In [24] statistically significant correlation between VLHR and nasalance was shown ($p < 0.01$). In another study by Dodderi et al., recordings of three sustained vowels were taken before and after the completion of cleft palate surgery for thirty children. The result showed a significant decrease in VLHR after completed surgery [6].

**Mel-spectrum** While not a feature in strict terms, the Mel spectrogram has become a fundamental tool for deep learning tasks involving speech and music analysis. It is based on the Mel scale, which is a frequency transform purposed to equalize listening perceptions for different frequencies. A human ear can easily hear the difference between 500 Hz and 1000 Hz, but not 10000 Hz and 10500 Hz, even though the difference is 500 Hz in both cases. Therefore, the Mel scale is a perceptual pitch scale based on what listeners find to be in equal hearing distance [29]. The attempt to mimic how humans hear and perceive audio is why the Mel scale is useful and often applied in machine learning applications. The conversion between a frequency $f$ measured in Hz to Mel is:

$$m = 2595 \log \left(1 + \frac{f}{700}\right) \tag{2.7}$$

As such, it corresponds to a logarithmic transformation. The result is that lower frequencies (measured in Hz) have a larger distance between them in Mels compared to higher frequencies. This mirrors how the human ear perceives audio, as discussed above. To construct the Mel spectrograms, we first need to estimate the periodogram $P_m$ for the specific frame $m_i$ of interest from the STFT. $P_{m_i}(k) = \frac{1}{N} |X(k, m_i)|^2$. After that, a Mel-spaced filter-bank consisting typically of 40 triangular filters is applied to the estimated periodogram[35]. Each filter has a response of 1 at the center frequency corresponding to that particular filter, and filtering the periodogram through this bank corresponds to the conversion outlined in (2.7) . In Figure 2.6 the filter bank outline is plotted with $n = 40$ filters.

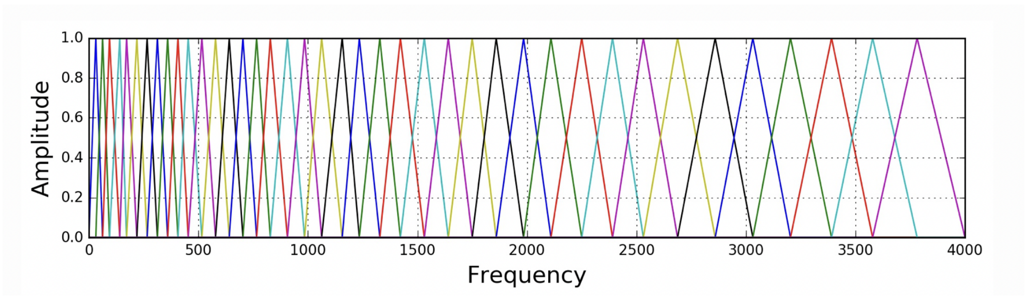

**Figure 2.6:** Mel filter-bank example, $n = 40$.

Returning to our example phrase "*Kicki kokar korv*" and plotting the Mel spectrogram for the adult and the child in Figure 2.7 shows that larger emphasis is now put on the lower frequency domain since this is where the human ear is better able to differentiate subtle differences.

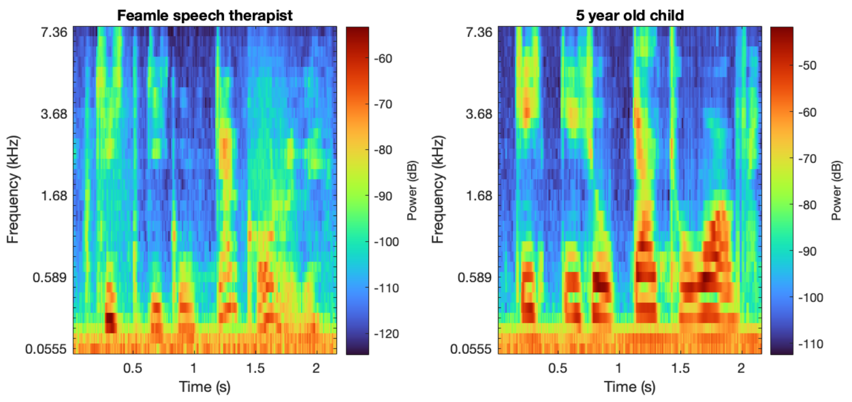

**Figure 2.7:** Mel spectrograms for the phrase "*Kicki kokar korv*".

**MFCC** To achieve a more compact representation of the Mel spectrum over time, it is common to transform the Mel spectrogram to its corresponding Mel-frequency cepstrum coefficients (MFCC). This is achieved by applying a discrete cosine transform to the Mel spectrum for a specific time frame. In this way, we construct a "spectrum of a spectrum" which indicates which parts of the Mel spectrum are most active for a specific frame but in more compact notation than the entire Mel spectrogram. By applying the DCT to more frequency bins, the amount of MFCCs can be adjusted, and a common choice is to extract 13 coefficients [33]. MFCCs features are rooted in the human ear's critical bandwidth for hearing, as briefly discussed above. These coefficients can therefore be regarded as an artificial implementation of the human ear's working principle. For this reason, MFCCs are utilized in many speech processing applications, for example it has been used for speaker recognition and other general pattern recognition when working with the human voice [7].

**Harmonic ratio** The harmonic ratio (HR) compares the energy contents in the tonal part of an audio signal to the total energy contained in the signal. Therefore it can be applied as a discriminant of voiced and unvoiced speech since the former is expected to contain more harmonic structures as discussed in section 2.1. The HR estimation is based on the normalized autocorrelation for the frame of interest [48]:

$$\Gamma(m) = \frac{\sum_{n=1}^{N} s(n)s(n-m)}{\sqrt{\sum_{n=1}^{N} s(n)^2 \sum_{n=0}^{N} s(n-m)^2}} \text{ for } 1 \leq m \leq M \tag{2.8}$$

$s$ is the time-domain representation of the frame with $N$ data elements, and $M$ is the maximum lag in the calculation. An initial estimate of HR is then given as

$$\text{HR} = M_0 \leq m \leq M^{\{\Gamma(m)\}} \tag{2.9}$$

where $M_0$ is a pre-determined lower threshold for the minimum lag search range. The HR estimate is then improved using parabolic interpolation.

**Spectral entropy** To measure the peakiness of the spectrum, a common feature to use is the Spectral entropy (SE). Similarly to HR, it has successfully been used to distinguish between voiced and unvoiced speech regions. Since entropy measures disorder, regions of voiced speech will, on average, exhibit lower levels [49]. SE is estimated as

$$\mathrm{SE} = \frac{-\sum_{k=b_1}^{b_2} s_k \log(s_k)}{\log(b_2 - b_1)} \tag{2.10}$$

where$f_k$ is the frequency in Hz corresponding to bin $k$, $s_k$ is the spectral value at $k$, $b_1$ and $b_2$ are band edges. The metric can be evaluated both on the Mel spectrum and the normal spectrum.

## 2.3 Source-filter Model

To approach the human vocal production system analytically, a commonly used framework is the source-filter model, which describes the voice output as the combination of a sound source (the vocal fold vibrations) and a linear acoustic filter (the vocal tracts and the articulators). Generally, a filter is a construction that allows certain objects or features to pass through it while the filter blocks other objects or features. An acoustic feature filter attenuates frequencies within specific ranges while letting other frequency ranges pass through unaltered. A simple example is a low-pass filter which lets frequencies below a certain threshold pass through it while higher frequencies are damped. Filtering of the discrete-time signal through a finite impulse response (FIR) filter can be expressed as a discrete convolution

$$\begin{aligned} y[n] &= b_0 x[n] + b_1 x[n-1] + \cdots + b_N x[n-N] \\ &= \sum_{i=0}^{N} b_i \cdot x[n-i] \end{aligned} \tag{2.11}$$

where $x[n]$ is the input signal, $y[n]$ is the output signal and $b_i$ are the coefficients representing the impulse response of the FIR filter $h[n]$ of order $N$

$$h[n] = \sum_{i=0}^{N} b_i \cdot \delta[n-i] = \begin{cases} b_n & 0 \le n \le N \\ 0 & \text{otherwise.} \end{cases} \tag{2.12}$$

A DFT of (2.11) and applying the convolution theorem yields

$$\underbrace{\mathcal{F}\{x * h\}}_{Y(\omega)} = \underbrace{\mathcal{F}\{x\}}_{X(\omega)} \cdot \underbrace{\mathcal{F}\{h\}}_{H(\omega)} \quad \text{and} \quad y[n] = x[n] * h[n] = \mathcal{F}^{-1}\{X(\omega) \cdot H(\omega)\} \tag{2.13}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denotes the DFT and its inverse, respectively. As such $H(\omega)$ is the complex-valued frequency response of the FIR filter.

A Fourier series define the response as

$$H_{2\pi}(\omega) \triangleq \sum_{n=-\infty}^{\infty} h[n] \cdot \left(e^{i\omega}\right)^{-n} = \sum_{n=0}^{N} b_n \cdot \left(e^{i\omega}\right)^{-n} \tag{2.14}$$

Here the subscript refers to the $2\pi$-periodicity of the function, and $\omega$ has the unit radians per sample. If the signal $x[n]$ is sampled at a known sampling rate $f_s$ we can substitute $\omega = 2\pi f / f_s$ and thereby change the frequency unit $f$ to cycles per seconds, also known as hertz (Hz).

The generation of voiced speech in the context of the source-filter model assumes that the vibrations of the vocal fold generate the input signal $x[n]$ in the form of a wave. Then, its spectrum predominantly contains energy at the fundamental frequency $f_0$ and its integer multiples (overtones or harmonics). The vocal tract and the articulators act as a filter on this input signal and thus modify the source into the resulting voice output signal $y[n]$. This filter is a complex system consisting of a number of different resonators, each with its own resonant frequency bandwidth, which can be modified by changing the shape of the resonator.

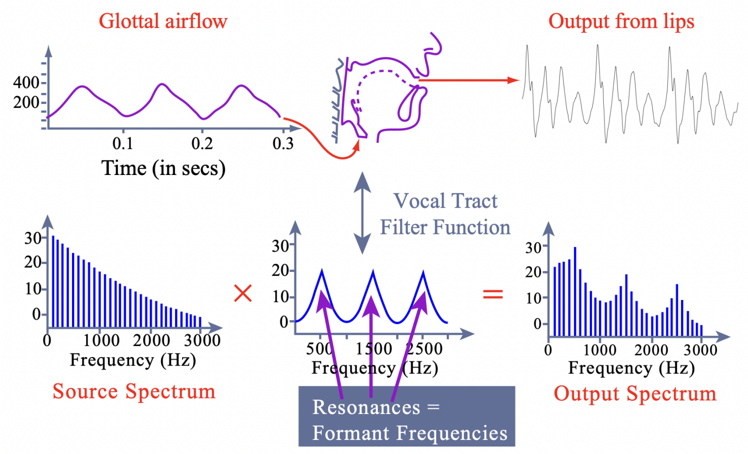

**Figure 2.8:** Source-filter model input-output transformation [9].

As such, the underlying idea of applying the source-filter model in the context of speech attribute classification is the proposition that the recorded speech output signal can be used to decode attributes of the vocal tract and articulators' configuration.

## 2.4 Cleft Lip and Palate speech

CLP conditions are a result of abnormal fetal during pregnancy between the 5th and 12th week, where different parts of the face are supposed to join. Speech development in CLP children is severely hampered before a surgical procedure can be performed to resolve the condition, after which there exists a great individual variation in how the child develops speaking skills. Four main factors usually decide how well the child can catch up with its peers. Namely, the type and severity of CLP, the timing and type of surgery, velopharyngeal function, and the availability of quality professional speech therapy.

CLP speech can be distinguished from regular speech and characterized by a few typical observations. Hypernasality is one of these, and it is the result of an open passage to the nasal cavity. Air that is supposed to flow out through the mouth instead resonates in the nasal cavity and flows out through the nose. Hypernasal speech is especially evident during voiced vowel pronunciation. An open passage also prevents the build-up of air pressure in the vocal tract because of the air leakage. This prevents hard pronunciations of certain consonants and makes them softer, and this difference can be observed through more energy content in lower frequency bands. The VLHR feature described above can capture this effect. There are also other pronunciation errors, mainly if the phrase is articulated in front of or behind the pharynx, meaning that certain plosives and fricatives articulated in front appear different. For example, a dental */d/* becomes a velar */g/*. Articulation behind the pharynx is, on the other hand, exemplified by fricative pronunciation in the back of the throat. These differences can be quantified with the help of the formant feature. If all the properties can be captured by relevant features or the indicative sounds themselves filtered out, good network results should be obtainable [13].

# 3 Deep Learning and Pattern Recognition

In the last decade, deep learning methods have risen to prominence in many research fields, given the wider availability of data and computing power. Applications usually involve predicting or classifying data for a certain end-use. A typical problem can be the classification of handwritten numbers from a picture into their respective categories. In this chapter, a brief background to deep learning and a justification for its use will be given.

## 3.1 Justifying the use of Deep Learning

While deep learning has proven to be a powerful tool in many applications, the case still has to be made for why it is a good option to classify speech for children with cleft palate. Two arguments will be presented for why such is the case.

Automating classification would, as stated previously, free up both time and resources for speech pathologists. By providing a sound file, a trained neural network can immediately classify the speech level. This has the potential to provide an unbiased and reliable classifier that can hopefully act as a complement to a human professional. Deep learning, therefore, provides a robust method to automate the entire process. Secondly, there is data available to train a network with several sound recordings of children speaking that can be utilized for this purpose. The availability of data makes a deep learning approach possible.

Several close examples have been successful or moderately so with deep learning applied to speech. Looking at these papers inspires and further strengthens the case that training a network can yield satisfactory results. A few notable examples are:

1. Graves, Schmidhuber. *Framewise Phoneme Classification with Bidirectional LSTM Networks*, 2005. In this paper, a 73.2 % testing accuracy is achieved for their stated objective [11].
2. Zhant, et al. *Automatic hypernasality grade assessment in cleft palate speech based on the spectral envelope method*, 2018. In this case, single utterances from children are classified with an accuracy range of 83.86% to 97.47% on Mandarin speakers [44].
3. Mathad, et al. *Deep Learning Based Predictions of Hypernasality for Clinical Applications*, 2020. Here results for four classes of hypernasality are classified correctly up in the 70%-75% range [25].

4. MathWorks has a used case Speech Emotion Recognition example in which they use audio recordings to classify emotions using LSTM network architecture. Results also range in the 70% range [50].

With these papers indicating good results with deep learning solutions, the availability of data, and the desire to automate the process, the case is strong for attempting such a solution.

## 3.2 Artificial Neural Networks

Deep learning relies on the availability of large data sets to discover hidden patterns in the data to learn the overall information structure that can differentiate data points or predict a future data point given a certain input.

At the heart of this process lies the Artificial Neural Network or ANNs for short. Its' structure closely resembles the neurons in our brain, where multiple connections exist between neurons to pass on information. Each connection has a weight assigned to it, indicating how important the information flowing through is for the classification or prediction. When the network is trained, it is these weights that are optimized.

A neural net is composed of several layers, where each layer has a specific function to improve overall performance. Some layers are good for image recognition, while others are more suited towards handling time series data. The goal is to find the right combination of layers to build a successful model. In this section we briefly cover some of the most frequently utilized layers when establishing an ANN architecture.

**The Fully Connected Layer** This layer is the workhorse of many learning applications and is featured in most network structures. Its' special ability is that it does not assume anything about the input (for example, having images as input) and is thus structure agnostic. According to the Universal Approximation Theorem, this layer can learn almost anything given sufficient computational power and data. It is therefore featured in many broad and varying applications as its strength is its generality. However, this comes at the cost of more specialized layers that can usually outperform the fully connected layer given the right application.

Looking at the mathematical representation of this layer, where $x, y \in R^m$ are the input and output vectors, respectively, the output can be represented in the following way:

$$y_i = \sigma(x_1 w_1 + ... + x_m w_m) \tag{3.1}$$

where $\sigma$ is a nonlinear activation function and $w$ the weights assigned to each neuron in the layer. These layers can also be stacked on top of each other, which goes by the term deep learning. A neural net can have different shapes, it can be built wide with fewer but larger layers, or deeper with more layers but fewer neurons in each. Even though there is a lack of theoretical evidence for the deeper networks to perform better, they have beaten the wider options many times. There is empirical evidence for this to be true, however, it cannot be thoroughly explained. The belief is that deeper networks can learn more complex relationships in the data and that they consequently perform better. But this is not true in all cases, and there are multiple examples of deep networks performing poorly.

On the other hand, there is theoretical evidence for a fully connected network to have strong universal approximation properties, meaning that it can learn almost anything in theory. In other words, this layer is a powerful tool that can detect patterns everywhere. The problem comes with the backpropagation part of the training when the weights are optimized, which is a complicated task. And optimization can be a difficult problem. The utilized optimization algorithms do not always find the global minimum, or sometimes any does not converge. This restricts the full potential of the fully connected layer. Nevertheless, it has still proven capable of solving difficult problems that could not be solved one decade ago [30].

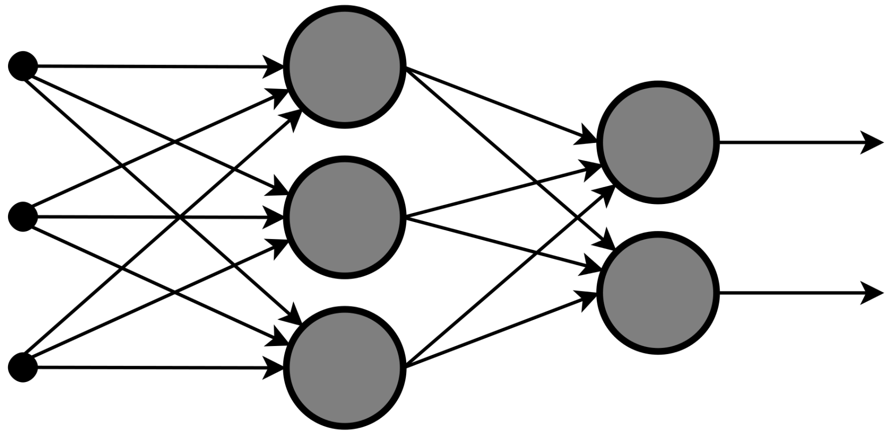

**Figure 3.1:** Structure of of multiple fully connected layers in an ANN [55].

**The Long-Short Term Memory Layer** This is a specialized recurrent neural network layer that can remember contextual information [20]. It has internal states that remember information about previous inputs. This functionality is used for prediction cycles in the network where a prediction is influenced by the current input and the remembered information, allowing the network to use contextual circumstances for each data point. The usefulness of remembering past inputs is quite apparent. If the data has sequential properties or a hierarchical decomposition, the LSTM network can utilize contextual information for better results. This layer is widely used in areas of language modeling, video analysis and speech processing [14]. The LSTM-layer can remember connections between inputs that are separated by more than 1000 discrete time steps. This is in opposition to a traditional recurrent neural network that could only handle sequences of 5-10 discrete time steps because of problems with exploding or vanishing gradients during backpropagation. In the case of the LSTM, error backflow is kept constant through the internal states by truncating the gradient at certain architecture-specific points and thus preventing the previous issues [16]. Each LSTM layer has memory blocks where the incoming information is processed and remembered, but for this to happen, the input must pass through two gates, an input and output gate that allows information to access and leave the memory block. The point of the input gate is to protect the memory block from irrelevant error information during backpropagation, and the output protects the network from irrelevant memories. The following equations guide input and output gates in the memory blocks.$s_c$ is the state value of cell $c$, i.e., its value after the input and forget gates (another way to remove redundant information) have been applied. $f$ is the squashing function of the gates, $\omega$ the weights to be trained regulating the incoming information [11]:

$$\text{Input gates update: } x_\iota = \sum_{j \in N} w_{\iota j} y_j(\tau - 1) + \sum_{c \in C} w_{lc} s_c(\tau - 1) \quad , y_\iota = f\left(x_\iota\right) \tag{3.2}$$

$$\text{Output gates update: } x_\omega = \sum_{j \in N} w_{\omega j} y_j(\tau - 1) + \sum_{c \in C} w_{\omega c} s_c(\tau) \quad , y_\omega = f\left(x_\omega\right) \tag{3.3}$$

While the LSTM only remembers past inputs, its close cousin the bi-directional LSTM layer can see into the future and use future information about inputs. The idea is to use two separate LSTM layers and connect them to a common output. By presenting the training sequence both forward and backward separately to the layers, the network remembers past and future information given any point in the input sequences [4]. Even if relying on future inputs for speech processing might seem like a violation of causality - how can humans base their understanding on something that has not been said yet? However, humans use this mechanism to some extent since words or utterances that make no sense are understood in light of future context. Sometimes we expect certain words to follow others [12].

**The Convolutional Layer** This is also a specialized type of layer, and it excels in applications of image recognition and computer vision. It relies on having image inputs and then passing the image through a filter that maps the results on a feature map. During training, it is the filter weights that are optimized. When the filter acts on the image input a discrete convolution between the filter and image pixels is calculated to form the output on the feature map. The promise is that these filters will help to detect outlines of edges and borders present in the image [34].

$$G[m,n] = (f * h)[m,n] = \sum_j \sum_k h[j,k] f[m-j, n-k] \tag{3.4}$$

where $G$ is the feature map, $f$ the image, and $h$ the filter. Stacking convolutional layers for computer vision applications has proven to be useful. Each layer is an extra step in the hierarchical decomposition of the data. The first layers can map lines and edges onto the feature map while later layers start to assemble the features together, forming shapes that later become faces, animals or objects of different sorts. During training, the network learns which features are most relevant for classification. This can be illustrated in a feature heat map where the hot areas in the picture below indicate important features [3].

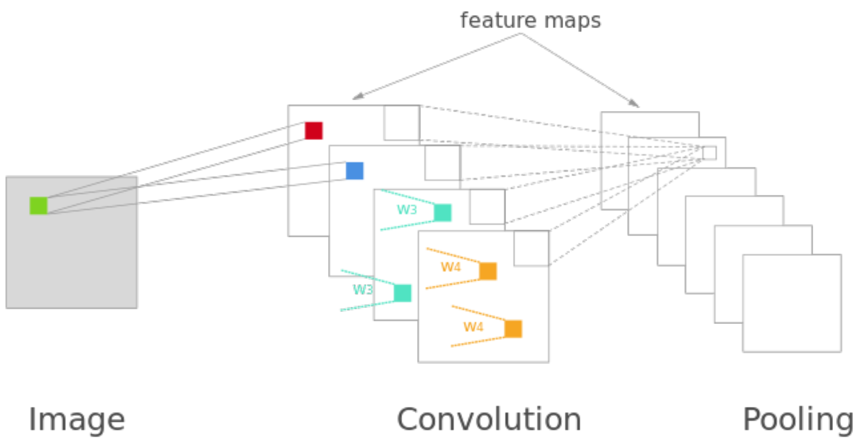

**Figure 3.2:** Visualization of the working procedure for a convolutional network [41].

**The Softmax Layer** To make a classification at the end of the neural network, there needs to be a rule for how the outputs from the last layers are converted into a classification. The softmax layer provides this rule by translating the final outputs into conditional class probabilities in the following manner:

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^{K} \exp(x_j)} \tag{3.5}$$

where $x_i$ is the $i$th class element in the output vector from the previous layer, each element representing a class. High softmax values can be interpreted as a high degree of confidence in the network's classification and vice versa. The softmax output with the largest probability score is selected, and the input to the network is classified accordingly.

## 3.3 Transfer Learning and Pre-Trained Networks

Not all deep learning applications require a network trained from scratch. A pre-trained network is a saved network already trained by someone else, typically on a huge data set and fully optimized for the best possible results. Several of them are available for public use and can be freely downloaded from the internet. Here are some examples of pre-trained networks:

1. **VGGish**: A network trained by Google to classify audio. Trained on 70 million Youtube videos [15].
2. **ResNet**: An image classification network that manages to keep both depth and a stable gradient [8].
3. **GoogleNet**: Google developed network used for image classification with 22 layers in total [32].

Pre-trained networks have proven to be useful when applied with Transfer Learning. The intuition behind Transfer Learning is that the pre-trained networks have been trained on large enough data sets to learn enough general features for the area you're working in. This way, knowledge from a related task can be used to improve the learning of a new task. The main advantage is that this network can be utilized to achieve good performance on small data sets by adding a few untrained layers at the end. The pre-trained network will create embeddings for the smaller data based on what it has learned from the bigger data set. These new outputs will then serve as inputs to the untrained layers in order to train them to convert these embeddings into relevant classifications or predictions. In other words, the pre-trained network is used to extract features [3]. In conclusion, pre-trained approaches decrease the need for a vast data set by relying on structures already specialized in discriminating between classes in the specific domain.

## 3.4 Hyperparameters and Training Options

All deep learning models have several parameters that affect their performance. These are known as hyperparameters and must be tuned adequately for the best results. Some hyperparameters in the layers mentioned above could be the number of neurons, filter size, and memory length for LSTMs. Good hyperparameter values can not be estimated during training, usually they must be decided through trial and error or a well-educated guess. Here are four of the most impactful and common hyperparameters [27].

1. **Model Architecture**: What layers should be used and how many? These are important decisions to make since they strongly affect model performance. The right combination of convolutional and fully connected layers can prove crucial for image classification
2. **Model Complexity**: How large should each layer be? A complex model can yield a higher accuracy at the cost of computing power. It can also overfit on your training set. A simpler model could instead fail to capture some of the more complex patterns in the data. Here the balance is everything. This is where the above-discussed hyperparameters for LSTMs and the other layers come in.
3. **Loss Function and Learning Rate**: The loss function calculates the cost of misclassification. Two examples are Least Squares and Cross-Entropy. Learning Rate, on the other hand, determines how large the step size should be during backpropagation. A smaller step size might be better at finding local minima and help the network to converge, while a larger step size might do the job faster.
4. **Optimizer**: What algorithm is used to minimize the loss during backpropagation? This is the algorithm that updates the weights during training.

## 3.5 The Problem of Overfitting

A large neural network can learn complicated patterns and their interactions. According to the Universal Convergence Theorem, it is possible to learn everything in a data set in theory. Problematically, this can result in overfitting when the network learns to memorize the data so well that it performs much worse on unseen data that was not part of the training set. This means that the network did not learn enough generality from the data set that should describe all available data. A training set is just a subset of all available data, and the hope is that the network can learn enough general features to perform well on new data. A small training set can also result in overfitting as there is less data for the network to memorize [1]. Here a few techniques to combat overfitting are presented:

1. **Simplifying the model**: A complex model will have an easier time memorizing the specific data in the set. Making a smaller model can prevent this and force the network to learn more general features. Simplification can be done by making the layers smaller, with fewer neurons, or removing some layers entirely.
2. **Early stopping**: It takes time for the network to reach a point where it starts to overtrain. Early iterations in training generally have a lesser generalization error. At some point, however, this starts to change. A typical pattern for overfitting is when the validation data error starts to increase significantly to levels above the training error in the training graph. Stopping the training just before this happens is a way to prevent overfitting.
3. **Generating more data**: By making the data set larger, it is harder for the network to memorize all data points. This is an alternative to reducing the model complexity. However, sometimes more data is simply unavailable. In that case, data augmentation might be an option. This is a way to artificially create more data by making small alterations in the original data set. For example, rotating or mirroring an image can do the trick. For voice applications, slightly changing the pitch can generate multiple different examples from the same sound file.
4. **Regularization**: This method alters the loss function by adding a penalty to it that increases with model complexity the absolute value of the weights, thus guiding the optimizer towards more balance in the weights. Two common penalties are the L1 and L2 regularizations, where L1 takes the sum of the weight absolutes. L2 squares the weights. Both have advantages and disadvantages. L1 is robust to outliers, while L2 is quite sensitive to them.
5. **Dropout layers**: This is a form of regularization layer that has the function to block randomly chosen neurons each iteration to prevent the layers from learning overly complex patterns in the data that lead to overfitting.
6. **K-fold Cross-Validation**: This method trains the network K times, each time switching what training and validation set it uses to avoid luck that can artificially boost results. Say, for example, that the validation data contains data points that, for some reason, are easier to classify for the network. This would give a higher validation accuracy than what might actually be the case

since the validation data does not represent a balanced sample from the sample population. To do a 10-fold cross-validation, 10 permutations of the training and validation set are created, with no overlap of data. The model is trained separately 10 times on these permutations, and a final accuracy is calculated as a mean from all runs. The variance between runs can be a measure of how much luck is a factor in the results.

# 4
# Data Set and Pre-Processing

In this chapter, the data set is introduced and analyzed. To convert audio files into digestible information for an ANN, a number of steps need to be performed. Data must be cleaned from irrelevant segments such as silence or adult speech and after that, features can be extracted. We discuss some of the most cumbersome aspects of this pre-processing pipeline.

## 4.1 Available Data

For this project 141 anonymized audio recordings were obtained containing conversations between children and a speech pathologist, recorded from one of their speech evaluation sessions. In these sessions, the children are asked to repeat or read certain phrases that could indicate a weakened velopharyngeal function to form a basis for assessment. The data was provided by Skånes University Hospital in Lund, resulting from a co-operational relationship with LTH. The length of the recordings varies between files, from less than a minute long up to ten minutes of audio. The children speaking are either 5 or 10 years old. Each recording is complemented with clinical data such as the type of CLP, surgeries performed and baseline data information (gender, age, etc.). Most importantly, the speech pathologist's rating of the velopharyngeal competence is provided. As discussed in section 1.1 this label is one of three categories. In Table 4.1 an overview of the data set is presented.

| Label | Definition | $n$ patients | % of patients |
|---|---|---|---|
| 1 | *Competent* | 59 | 41.84% |
| 2 | *Marginally incompetent* | 52 | 36.88% |
| 3 | *Incompetent* | 30 | 21.28% |
| | | 141 | 100.0% |

**Table 4.1:** Overview of the data set.

For ease of further reading, the categories will sometimes be referred to as Category 1, 2 and 3. One immediately striking point is the unbalanced nature of the data set, as there is a significantly fewer number of children in Category 3. Skewed distributions in the data need to be addressed with care in modeling stages to avoid bias in predictions. Furthermore, it is important to investigate and highlight other imbalances concerning the data, especially for attributes that are believed to have a direct effect on, e.g., the spectral contents of the recordings. The vocal cords and the vocal tract are longer for a 10-year-old than for a 5-year-old, and the produced voice is hence fuller and darker. Boys produce, in general, lowered pitch speech compared to girls, and the difference is more pronounced for 10-year-olds as the child approaches puberty. We plot the distributions of age and gender in the 3 categories below to the end of this reasoning.

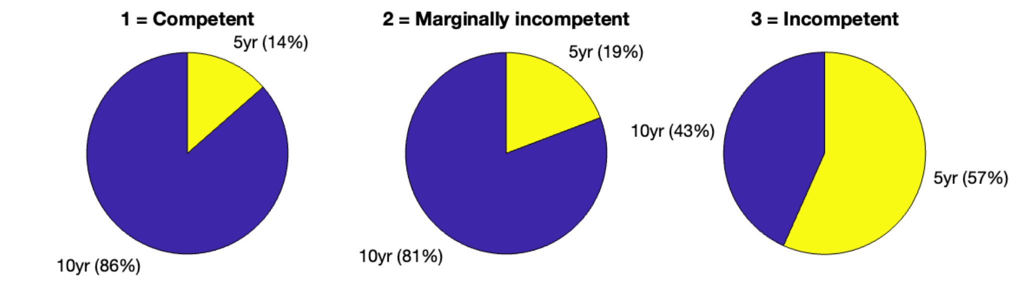

**Figure 4.1:** Age distribution in categories.

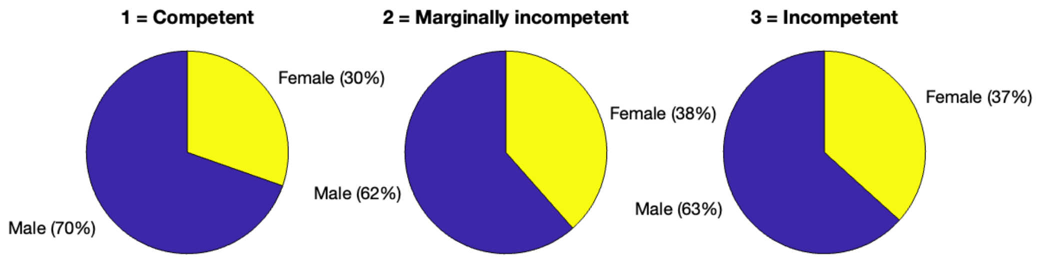

**Figure 4.2:** Gender distribution in categories.

From Figure 4.1 it is clear that Category 3 differs significantly from the other two with respect to age distribution. A majority of children in this category are 5 years old while in the other two categories 5 years old are in a clear minority. While this hints at satisfactory results from speech therapy (and possibly surgeries), it might cause problems in the network's training. Intuitively, if not fed age information for a recording, the network could regress to classifying high-frequency speech as Category 3 as it is unable compensate for the fact that 5-year-olds usually have a higher pitch. To remedy this, the decision was made to include age in the feature vectors given as input to the network. Figure 4.2 does not indicate bias in the input data for gender, at least not to the same degree as for age. There is a slight overweight of boys in the data, but since this imbalance is present across all categories, it is not likely to cause structural issues in predictions. Nonetheless, including gender in the feature vectors might yield better results, especially for predictions on 10-year-old subjects. Furthermore, the decision was made not to include any clinical data pertaining, e.g., the type of CLP condition and performed surgeries. The effect of such factors on speech is undoubtedly an interesting subject of great importance to medical professionals, but it is not the scope of this learning task because of one key reason: The goal with the trained algorithm is to perform on the level of a highly trained speech pathologist, and the speech therapists rate the patients based on the speech contents in the recording and *not* based on clinical data. An argument can be made that speech pathologists might be biased with respect to these factors, but since such a bias is an undesirable aspect of an objective algorithm.

## 4.2 Pre-processing of recordings

As a first step, the raw `.wav` audio files containing the recordings were loaded onto a PC. In order for the data to be properly processed, it has to be normalized. Audio can be recorded in different manners, and there are structural variations like sampling rate, the distance between speaker and microphone that affect volume. Therefore, each recording is resampled to a 16 kHz sampling rate and the signal amplitude normalized between -1 and 1 to mitigate variations due to volume in microphone placement. To better understand the structure of a speech therapy session, we listened back to a number of recordings and came to two initial conclusions:

1. Generally, the recordings consist of speech content from two persons - the speech pathologist and the child. This, however, differs from recording to recording. For example, in some sessions with 10-year-old children, they are asked to read back sentences from SVANTE themselves, and as such, the recordings consist exclusively of child speech. In other recordings, the session is a conversation between the speech pathologist and the child. Ideally, features should only be extracted for segments of the recording containing child speech, not silent regions or speech from the speech pathologist.

2. As discussed earlier, a typical analysis frame for speech data to ensure WSS is between 20-30 ms and for a 10-minute long recording for this translate to approximately 24.000 frames. It is reasonable to assume that not all of these frames contain information directly influencing the rating of the child's velopharyngeal competence, and as such, they can to some degree be considered noise in the learning process. Filtering out such information noisy frames and only feeding the network with frames containing, on average, more relevant information can therefore enhance learning. This is a complex issue since excluding frames from the scope of analysis means that the input data set is reduced, and if not done accurately, there is a risk that relevant frames are left out.

We constructed two separate pipelines to combat these challenges, which we will now cover in more detail.

**Speaker Diaritization** This is the task of partitioning a stream of audio into homogeneous temporal segments according to speaker identity, which in this case translates to separating child utterances from speech therapist by audio frame. To solve this challenge, we tested multiple naive methods, such as, e.g., estimating the pitch throughout the recording and applying different means of *clustering* to identify two distinct categories, hopefully pertaining to the child and the adult. Unfortunately, this result was not satisfactory as many frames are wrongly classified as belonging to the child. We suspect that a major contributor to this is that the speech therapist often speaks in a "children's voice" when talking to a child, thus talking in a pitch range outside their usual domain. The majority of speech therapists in the data set are women who tend to have a higher pitch than men, adding to the separation difficulties.

With the use of SVANTE, similar phonemes and phrases are repeated when comparing the recordings. However, since the patient, in this case, is a child, it is not as straightforward as simply instructing them to a list from top to bottom. Speakers speak as they please, especially the 5-year old's. One approach would be to manually label all speaking segments for all files. However, this was deemed too time-consuming, and since it does not help with the end goal of a fully automated process, this approach was discarded. Another option would be to ignore the issue of speaker diarization entirely and feed the audio file in its entirety for further feature extracting, hoping that the network itself might find the most useful sound segments for classification. However, this approach was deemed unlikely to work given the small amount of data available. Hence, our conclusion is that the best option, given the small data sample, is to ensure that all input is of the highest possible quality, believing that a good input with features that highlight CLP speech should help the network learn the right patterns for this classification task.

Speaker diarization is a research field in itself with many ongoing developments [37]. Most publicly available solutions are either divided into labeled or unlabeled diarization, and the results themselves vary greatly between applications and underlying data. Therefore it is difficult to find a pre-trained network fitted for our end use. A few of them, like the spectral clustering method described in the reference above, were implemented in `Python` and applied to the data set [38]. Results were however unsatisfactory as the two classes (child and adult) had too many miss classifications in them, almost every other segment was wrongly classified. spectral clustering is considered to be a state of the art algorithm, building upon the i-vector method, produced by a Google research team in 2019. After attempting to extract features and pre-process them, the algorithm used the unsupervised k-means clustering method, utilizing a cosine distance metric to create clusters. A reason it might not have worked so well is that the features extracted are better suited to distinguish speakers in real conversations, with longer speaking times. It was trained on the NIST SRE 2000 CALLHOME data set, which contains phone call conversations. A phone call is much more structured compared to the more chaotic recordings in our data where speakers often overlap or speak in shorter utterances. Eventually another solution was found to work the data set. This method utilizes transfer learning, which after some tuning to our specific needs worked well enough to proceed. It based on the pyAnnote library in `Python` which is built on `pyTorch`. It has a customizable pipeline for speaker diarization This pipeline provides a full end-to-end speaker diarization of an audio recording in raw waveform input [2]. It consists of a few steps including voice activity detection, speaker embedding and clustering. In the following section, a brief presentation of the key aspects of the pipeline is covered.

**pyAnnote** Speaker diarization can be regarded as a sequence labeling task. Let $X = [x_1, x_2, x_3, \ldots, x_N]$ be a sequence of feature vectors extracted from 20ms frames, each element representing a single frame or a sequence of frames depending on the chosen configuration. The output should be a corresponding sequence of labels $Y = [y_1, y_2, y_3, \ldots, y_N]$. Not all frames of an audio recording can be used for analysis. Except for the speakers' voices, there is also silence in the recording. Think about the small pauses between speakers in a conversation or the silence sometimes present at the start and stop of the audio recording. These must be removed to keep the data quality high, and typically each recording is segmented into short audio frames. In this case, each frame is 20 ms, and frame-wise classification can be made to determine what is silence and speech. The silent frames are then discarded as there is no use for them. This procedure is often referred to as *voice activity detection* in literature.

To do this, an algorithm proposed by Lavechin et al. is employed to detect voiced frames [31]. In short, they utilize transfer learning by inputting raw waveform audio into a pre-trained network called sincNet and then adding their own layers. Consequently, the outputs are frames containing speech filled with the voices of the speakers. A brief word on sincNet, the network's original purpose was to recognize speakers with the help of convolutional layers. The idea in the paper is to encourage the first convolutional layer to discover more features by placing constraints on the filters by letting them convolve with a set of parametrized sinc functions that implement bandpass filters. Ideally, this can better highlight features like formants and make patterns more apparent for the network.

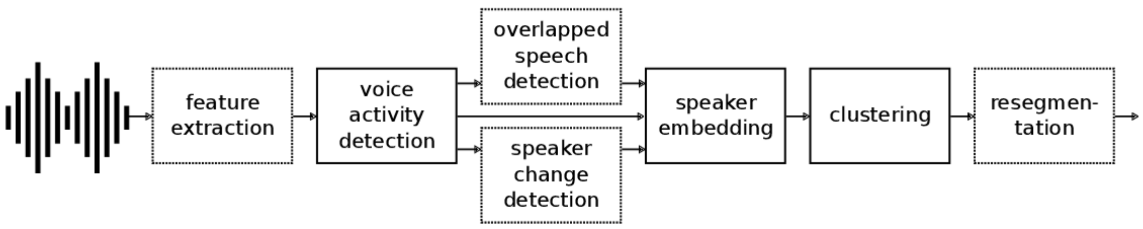

**Figure 4.3:** pyAnnote pipeline for speaker diarization.

The output from sincNet yields embeddings of the audio frame that are then fed into two LSTM layers, followed by three fully connected ones. In other words, sincNet is a feature extractor, and the entire network is trained to classify a frame as "*speech*" or "*non-speech*" [23]. The next steps in the pipeline are detecting speaker change and overlapped speech frames. PyAnnote once again utilizes pre-trained networks to achieve this. A pre-trained network consisting of BiLSTM layers trained on Broadcast TV data classifies frames where it detects a change to detect speaker change [43]. For speech overlap, a pre-trained model also classifies frames where it detects more than one speaker talking. Together with this information, speaker embeddings are extracted from the frames, and they are clustered to form a cluster for each speaker. Clustering is done by the k-means algorithm; a metric learning approach used to calculate distances between embeddings. In this pipeline, the cosine distance metric was used. The cosine distance metric is an orientational measure, meaning that it is the angle between two vectors $A$ and $B$. A small value indicates a similar orientation in the vector space, however, it says nothing about the magnitude of these vectors.

$$\text{similarity } = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4.1}$$

When the clustering has converged, the question is how to determine which cluster indicates which speaker, pyAnnote only gathers frames into clusters with no specification of who is speaking. It does, however, re-segment consequent frames and gives time points for all re-segmented audio frames in the same cluster. The issue now is to label each cluster per file as "*child*" or *"speech therapist"*.

**pyAnnote Post-procesing** Firstly looking at the clustering results, a few audio files managed to converge into more than two clusters, even though the file only has two people talking. These were checked manually to make sure no errors make it past the line and because it does not take too long to check. A simple pitch averaging estimation was made for the two clusters in each file for the remaining clusters, classifying the higher average as a child, keeping it, and discarding the rest. Results were manually double-checked and assessed as satisfactory by just listening. There are some segments with the speech therapist uttering a word or two, however, this was deemed as an impossibility to avoid. pyAnnote clearly had the best results of all methods tested, and listening to the remaining recording confirmed this.

**Identifying information-dense frames** As discussed in section 2.1 it is common to classify speech as either being voiced or unvoiced. The level of velopharyngeal competence is likely to impact the quality of voiced speech to a larger extent when compared to unvoiced speech since the former involves more resonances in the vocal tract. Hypernasality is, as stated earlier, a common feature in CLP children and can therefore be used to assess the severity of cleft lip or cleft palate speech. Hypernasality has been shown to be easier detected on voiced speech compared to unvoiced speech [25]. With this in mind, we constructed a copy of the data set, which is reduced by dropping segments believed to consist of mostly unvoiced speech. Practically this was achieved by analyzing each file in 0.2 s windows and dropping segments below a certain threshold percentile in the empirical distribution of Harmonic Ratio for that specific file. To evaluate the result and set a proper threshold, the reduced recordings were played back until satisfactory results were achieved. A good trade-off threshold was found to be around 75%, i.e., keeping the 25% segments with the highest level of Harmonic Ratio and Spectral Entropy.

# 5
# Training and Validation

With the data pre-processed, the natural next step is to build a suitable neural network that can classify audio recordings of CLP children. In this chapter, justifications will be provided for what features were selected, what layers are suitable for the neural network architecture, and what hyperparameters gave the best results. Moreover, descriptions of how the best model was found and the iterative process in reaching it are provided.

## 5.1 Data Partitioning and Labeling

Standard practice in deep learning involves splitting all data into different data sets. In this project, the data is split 70/30 to form separate training and validation data sets. This might seem like a smaller percentage allocated towards training than usual, but since the total data amount is quite limited, the need to factor out luck between for a specific run becomes apparent. Thus the decision was made to allocate more data to validation by using a 70/30 split instead of 80/20.

When partitioning the audio file into frames for feature extraction, it is of utmost importance that the training and validation sets do not both contain frames from the same recording, as this could artificially inflate results. If frames from the same recording make it into both data sets, the network might learn to recognize speakers instead of CLP utterances. Therefore, results for new data will be poor as the new data has no frames in the training data. Full audio files are placed in either data set, and consequent processing is done on both sets separately to prevent this from happening.

## 5.2 Searching for the Best Model

In order to compare two model instances with each other, a metric that measures performance is needed. Model accuracy, defined as correctly classified frames expressed in percentages of total classifications, is used here to compare runs between two instances. This metric can compare runs within the same model, i.e., when determining if an added feature or layer gave better results or to compare performance between two completely different model architectures. An example would be to compare a convolutional with an LSTM based model.

$$\text{Accuracy} = \frac{\#\text{ Correct predictions}}{\#\text{ Total predictions}} \tag{5.1}$$

The choice of features is critical for a good model. The goal is to include features pertaining to CLP while at the same time keeping the model relatively small in parameter space according to the "*Keep it simple, stupid*" principle. However, the selection process also involves trial and error. An educated guess can form initial features, but it should be combined with testing several permutations of other options as well. In the beginning, only a few are tested for a benchmark with 10-fold cross-validation. Subsequent features are then added or removed and compared to the first benchmark. When results no longer improve, the next step is to move on to optimize layers.

Deciding what layers can best translate the features into a high classification accuracy is the second step in the model search. In chapter 3, a few layers and their uses were discussed. Based on the theory, initial layers can be set up to give the initial benchmark, and subsequent additions can then be compared against it. When optimizing layers, freezing the features is a good idea to make sure that only layer changes are responsible for any performance lift. The point is to start small, with few features with few models, and then iteratively build upon them.

The last step to finalize the model is to select the best training options. This is achieved by trial and error. By freezing the features used, network architecture, and parameters, each training option can then be optimized one at a time. For example, several 10-fold cross-validation runs were made with different learning rates to obtain the best learning rate. The best result is chosen before moving on to the next hyperparameter, like the batch size, for example. Here are the most common training options that were tuned in this thesis:

1. **Number of Epochs** is by definition how many times the entire data set is passed through the network, essentially a setting for how long the network is to be trained. One epoch represents a full pass of the entire training set through the network.
2. **Learning Rate** is the setting for the optimizer used during backpropagation and specifically determines how large the iterative step sizes should be when the loss function is minimized. A larger learning rate is standard for the first epochs as the potential gains in performance are large, given a random initialization in the weights. After a few epochs, decreasing the learning rate might make it easier for the network to find a local minima. Since the gains in network performance start to thin out, a smaller step size could help here. Learning rate decay is another factor that can be set, and it just determines how quickly the learning rate should decay.
3. **Mini Batch Size** is how much data the network should see before weights are updated through backpropagation. A full batch, meaning all data in the training set, will be less prone to randomness and better manage to update the weights since the gradient will have a larger chance of pointing to the global minima and training the network for optimal decision points. However, this approach is computationally expensive as all data must be kept in memory. Furthermore, it takes a longer time to train the network this way. Having mini-batches alleviates the problems with time and computational expense but opens the door to randomness. This usually results in a more random

training process. Imagine a mini-batch containing the extremes of a data set. This will optimize the weights in the wrong direction, thus creating a more randomized training process. But if data is a good representation of the overall population, this will not be an issue. The network still has good chances to converge to a decision point.

To know if new network settings regularly outperform the old ones, it is vital to use cross-validation for robust modeling. Since the data set is small, luck during splits into training and validation can truly have an adverse effect. Thus, by generating 10 different permutations of train and validation for the new settings, a good indication in the results has a higher likelihood of being dependent on the settings and not on which data is validated. A model can be optimized by starting to optimize features, followed by layers, and finally, training options. The point is to freeze all else except for one feature, layer, or option in the next code run to isolate the potential effects of one change. This procedure is repeated, only now the first benchmark is from the resulting model in the entire previous trial and error iteration. This procedure provides a systematic working approach to find a candidate for the best model once the data has been pre-processed in a chosen manner.

The 10-fold cross-validation will yield an accuracy score for frame classification per class for each of the ten different runs. These results can be averaged for a robust estimate of the model performance. Furthermore, the standard deviation can be calculated as a measurement of model consistency. A large standard deviation signifies an unstable model that will be difficult to use consistently for real-life applications, while the opposite is true for low standard deviations. The main guiding evaluation metric will be the overall accuracy for frame-wise classification. Considerations will, however, also be given to category-wise accuracy and standard deviations to determine a possible improvement. The aim is to construct a model that can classify all categories equally well. However, the challenge in this is recognized given the unbalanced validation data.

## 5.3 Network Architecture Candidates

By studying earlier literature, it is clear that several network architectures have reached good results on similar tasks. Therefore, three completely different approaches will be attempted and evaluated. By doing this, the most likely models to work well will be tested. The models themselves are clearly defined, together with training options in the appendix. Instead, in this section, model approaches are justified and eventual model-specific pre-processing detailed.

### 5.3.1 The BiLSTM Model

Considering that speech contains significant amounts of contextual information, it is believed that the LSTM layer is a suitable building block for a CLP classifier. Furthermore, previous related research also utilized the layer in similar projects with good success. Therefore an initial network architecture with LSTM layers is a good starting point. Moreover, the type of LSTM chosen for this is the BiLSTM that can take in both past and future information about an input. As with the feature selection case, the final model is reached by comparing an initial model benchmark with trial and error methodology to obtain the best architecture. Starting out small and adding subsequent layers of different kinds to see if the benchmark can be beaten.

Network inputs are frames of 20ms, grouped in sequences of 20 frames per network input. Each sequence will have the same label as the file. Features are extracted per frame and normalized by the standard score (sometimes referred to as Z-score) method, subtracting the mean and dividing with the standard deviation for every feature present in the frame. Means and standard deviations are calculated on features from all available data to reflect the ground truth best.

Relevant initial features for a BiLSTM model would be VLHR, Formants, and MFCC since network performance is enhanced when features are engineered to better distinguish between classes.

### 5.3.2 The CNN Model

The Mel-spectrogram is a feature capturing the distribution of frequency contents in different frequency bands. It is known that CLP utterances can contain more energy in the lower frequency bands around 50-600 Hz [24]. This is visualized in the location of the formants. By letting convolutional layers map features from the Mel-spectrogram onto a feature map, the network can then utilize this discrepancy to find a useful pattern for classification. The input in this approach will therefore be a Mel-spectrogram of appropriate size. In this case, each Mel-spectrogram is 200ms long, believing that 200ms frames are long enough to contain at least one utterance signified by CLP. This increases the quality of the frames, as each frame will have relevant information. Listening to frames of this length, it is possible to hear short indicative utterances like "*bil*". Longer frames would possibly dilute relevant features, while shorter frames could miss them entirely.

### 5.3.3 T.L VGGish

The pre-trained network VGGish is a suitable candidate for transfer learning as it is trained on a huge audio data set. Therefore it should capture general audio features and can thus easily be adapted for the specific end-use in this thesis [15]. Because transfer learning produces good results for similar tasks, attempting to model this approach is plausible [3]. The point of transfer learning is to let the pre-trained network generate feature embeddings. For VGGish, the only input required is raw audio. The VGGish network calculates the Mel-spectrogram and passes the information through several CNN layers. Each mel-spectrogram is a window of

0.96 seconds and is given the same label as the file. The resulting output from the network is then fed into added BiLSTM and fully connected layers.

## 5.4 From Frame to File

Once frame-wise classification is completed and optimized for the best possible result, it needs to be re-segmented into a file-wise classification based on the frame results. There are a number of approaches to choose from when classifying a file, one of which is just a natural majority decision. The higher frame-wise accuracy achieved, the better this method will work as the correct class majority's probability increases. This method is highly reliant on a good performance when classifying frames. An imbalance in the data could make this metric biased. If the network has a bias towards a certain class, this will be reflected in the majority decision.

Another approach is to use the information available in the softmax layer output. The probabilities for a frame belonging to each class can be used in numerous ways. One option is to choose a threshold for when a frame classification is deemed certain, in this case, 70%. The idea is that many frames will have quite uncertain probability scores for each class, meaning that little information can be extracted from them. If a classification were to be made on maximum average probability scores for each category, these uncertain frames would pull down the average as they are quite many and low in probability scores. Therefore any averaging operation might be flooded with information from bad frames. By utilizing a threshold, these can be weeded out and further analysis made on high-quality classifications. The two methods of forming a classification decision can be summarized as

1. **Majority decision** A simple majority decision, the one with the most classifications wins.
2. **70% majority decision** Only frames with a 70% class probability are considered and a majority decision is made on these

Files are classified according to these metrics and the best measure selected moving forward.

## 5.5 Results

Results are presented, both frame-wise and file wise for each model. The used metric is total classification accuracy and its' standard deviation from 10-fold cross-validation. In the table below a summary of the results for the models are given and in the following sections, each model's performance is further analyzed more carefully.

| | Frame-wise | | File-wise | | |
|---|---|---|---|---|---|
| | Accuracy (%) | Std. (%) | Accuracy (%) | Std. (%) | Decision |
| BiLSTM | 52.92% | 3.44% | 56.05% | 7.23% | 70% Maj. |
| CNN | 74.25% | 1.69% | 89.76% | 3.41% | Majority |
| T.L VGGish | 55.39% | 3.73% | 57.67% | 6.83% | Majority |

**Table 5.1:** Result for the three network architectures.

Looking at the results, the CNN model clearly outperforms the others by both accuracy and standard deviation.

## 5.5.1 The BiLSTM Model

This model had the lowest frame-wise accuracy and therefore also performed the worst file-wise. The confusion charts illustrate the model's difficulty in distinguishing between Category 1 and 2 since most of the errors are clearly concentrated in the upper left quadrant. BiLSTM layers do, however, detect Category 3 much better. According to the model, the differences are smaller between the first two categories, making them harder to distinguish. As such, it would perform well as a binary classifier if the task was limited to distinguish Category 3 (i.e. incompetent speech) from the two other categories.

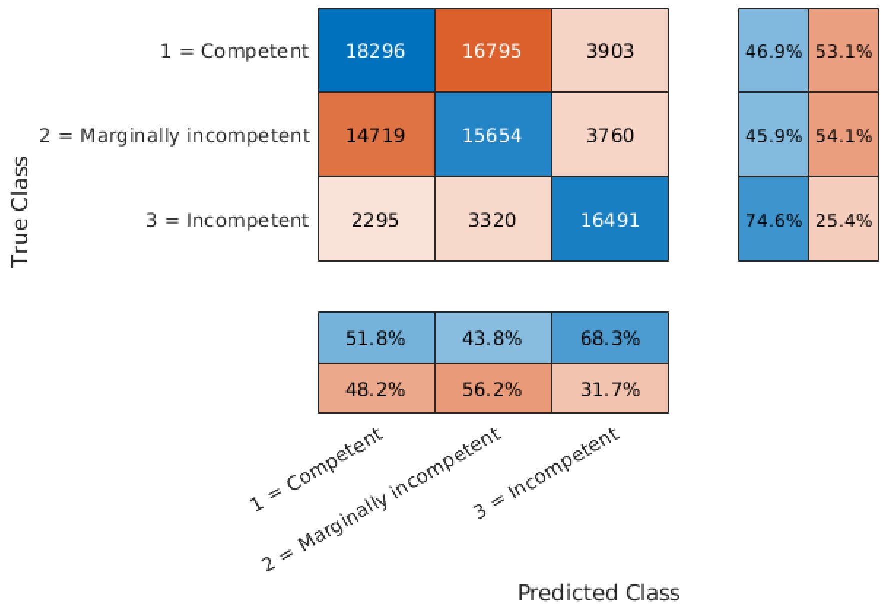

**Figure 5.1:** BiLSTM: Frame-wise confusion plot.

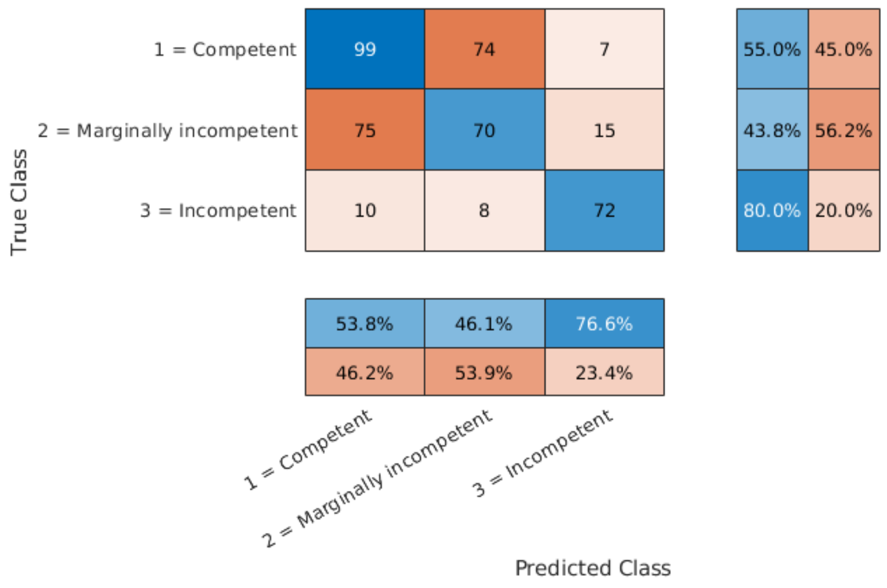

**Figure 5.2:** BiLSTM: File-wise confusion plot.

A number of different feature sets were tested in order to reach as high accuracy as possible. The most consistent result was given by the following configuration

| Selected features |
|---|
| MFCC |
| MFCC delta |
| Harmonic ratio |
| Spectral slope |
| Pitch |
| Age |
| Gender |
| First formant |
| Second formant |
| VLHR |

**Table 5.2:** Optimal feature set for BiLSTM model.

### 5.5.2 The CNN Model

By far the best model, the deep convolutional network accurately manages to classify between all three categories with little variation. The high frame-wise accuracy produces a similar result per file. Since all frames belonging to the same file are given the same label, high frame accuracy immediately translates into high file-wise accuracy. This effect is clearly seen with this model.

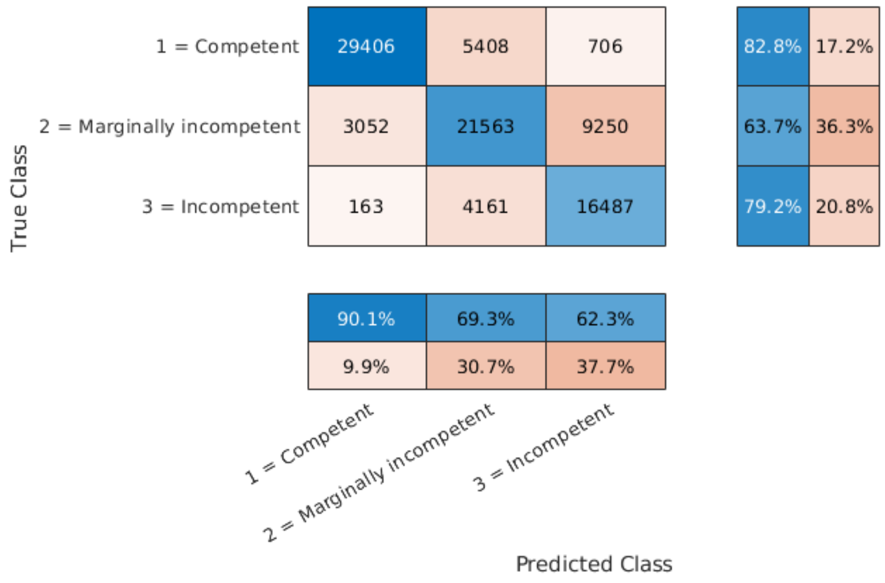

**Figure 5.3:** CNN: Frame-wise confusion plot

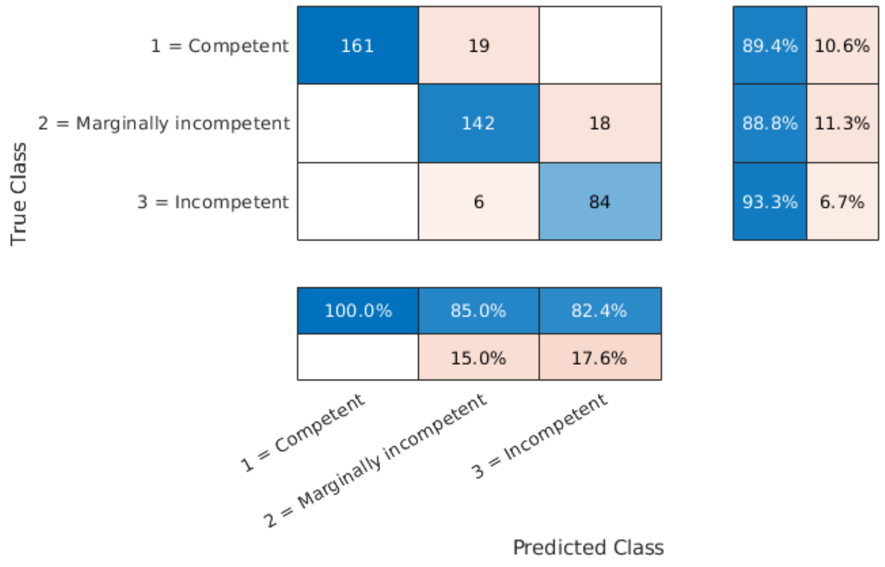

**Figure 5.4:** CNN: File-wise confusion plot.

### 5.5.3 T.L VGGish

This model performs similarly to the BiLSTM option explored earlier when looking at total classification accuracy and robustness. It should be noted that for Category 3, a 96.8% sensitivity on a file-wise basis, the highest result for all. Like the BiLSTM model, the factor hampering performance is the ability to distinguish between the first two categories. Like the BiLSTM, the focus for this model is to correctly classify Category 3.

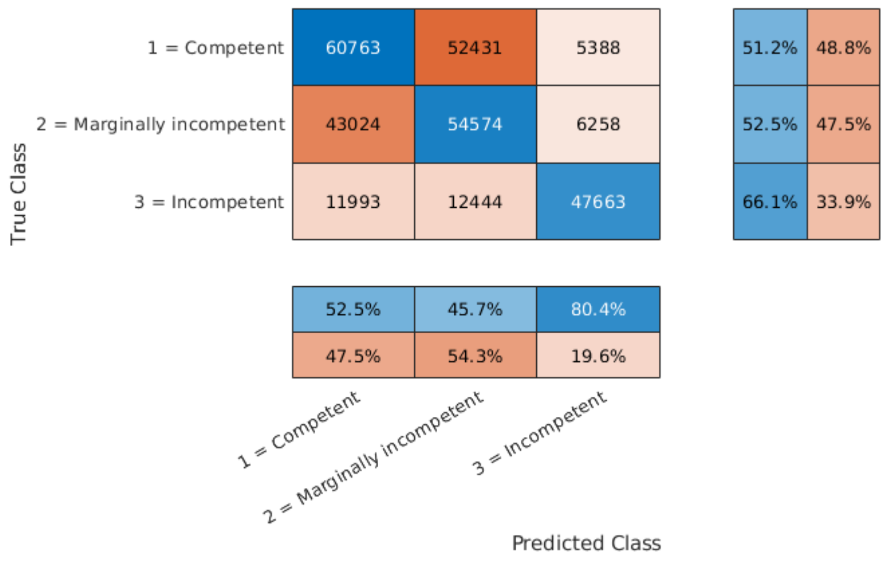

**Figure 5.5:** T.L VGGish: Frame-wise confusion plot.

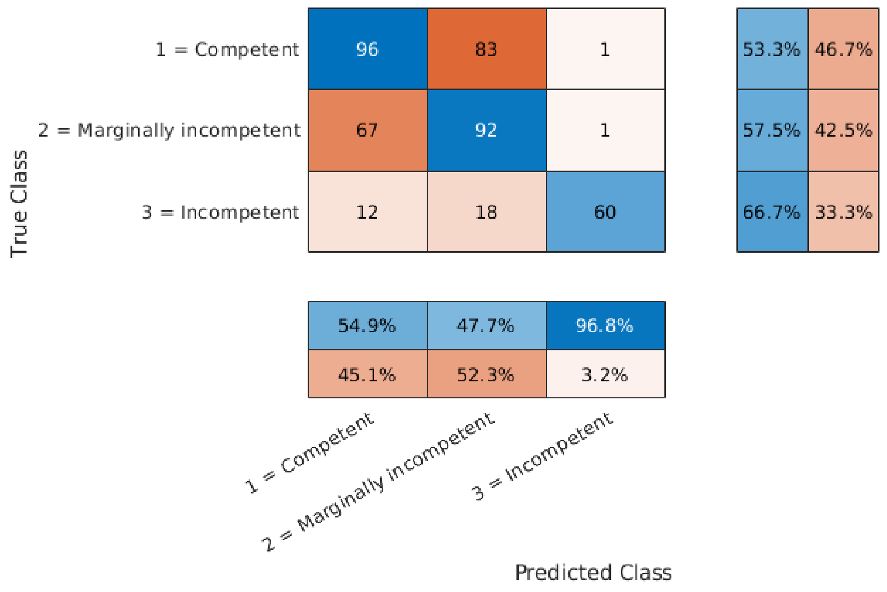

**Figure 5.6:** T.L VGGish: File-wise confusion plot.

# 6
# Discussion and Conclusion

In the final part of this thesis, the results given by the three models are discussed, and potential reasons behind them explored. Moreover, recommendations about future work are discussed. Potential steps to improve model performance and robustness are given.

## 6.1 Results Discussion

The previous chapter concluded that the CNN model had the best results and that both the BiLSTM and the T.L VGGish model performed worse but similarly.

**BiLSTM Model** This model had an accuracy of 52.92% when classifying frames. Work with the BiLSTM model was clearly the most time-consuming since iterations between different feature-sets was necessary. For feature extraction, there is also the aspect of choosing overlap and window length since different lengths can impact results. Despite so many options to tune, results over 53% were never obtained with 10-fold validation. The belief is that all options were exhausted to find better results. Therefore, it is plausible to believe that the data itself was simply not of sufficient quality to extract features from. One problem with labeling all frames with the file label is that CLP information-rich frames are diluted. Not all audio frames from a recording signify CLP speech. Speech pathologists listen after specific articulations to assess severity. The hope is for the network to do the same, which would be easier if frames were labeled differently in the same recording. However, since this is not the case, important frames are diluted among the rest, making it difficult for the network to learn which frames to concentrate on. It is instead trained to classify entire sequences containing both indicative and non-indicative frames with the same label. Furthermore, features are extracted frame-wise. Consequently, features like VLHR that detect hypernasality best on voiced speech are also extracted on unvoiced frames (despite the efforts outlined in section 4.2) and weighted equally as inputs. Frame-wise variations for VLHR and other features are therefore dependent on the underlying sound and less on CLP. The evidence for this train of thought is earlier research cited above. In those papers, the data was consistent and highly standardized with respect to phoneme spoken and recording setting. In other words, when dealing with a set of relatively non-unified recordings it is possible that the features, while well engineered, capture aspects other than velopharyngeal competence resulting in poor performance.

**CNN Model** Reaching a frame-wise accuracy of 74.25% by training on 0.2 seconds long Mel-spectrograms, this is the most promising model to develop further. The Mel-spectrogram feature can capture many of the distinguishing properties of CLP speech, like formant location and frequency distribution. When air from the vocal tract leaves through the nose, it affects the frequency distribution of the speech. This is more clear for some utterances but is present throughout the speech. This could likely explains the relative success of this model compared to the rest. Convolutional layers are strongly geared towards detecting patterns in images. Formant locations are clearly visible in the mel-spectrogram by humans. Therefore, it is not unreasonable to believe that a convolutional layer could detect more complex patterns responsible for the results.

**T.L VGGish Model** This model performed slightly better compared to the BiLSTM, with an accuracy of 55.39%. This model has fewer parameters to tune as the VGGish network does feature extraction from a raw audio input. Mostly, work was directed to choosing appropriate embedding sequence lengths and layer configurations for the layers following the VGGish output. This means that most network architectural aspects were tried. The feature used for the embeddings by the pre-trained network is the Mell-spectrogram, which has had good results with convolutional layers. Furthermore, looking at the structure of VGGish, it contains several convolutional layers and likens the CNN model structure in many ways. The poor performance can then only be traced to the training of the VGGish model. As mentioned previously, this network was trained on 70 million hours of YouTube videos that can contain all kinds of sounds. Therefore, this network is more geared towards distinguishing different types of sound, i.e., music, speech, noise, etc. There is a risk that the generated embeddings are too broad for the highly specialized end-use in this project. The similarity between CLP speakers in different classes is closer compared to the distance between music and speech. This fact could make the embeddings sparse with useful information, thus leaving most of the learning work for subsequent layers that resemble the BiLSTM model. This could explain the similarity in results. However, one thing to discuss is this model's ability to accurately classify true positives for the Category 3 of speakers. The difference between the third and two other categories is much greater, making the network act as a binary classifier. This effect is also seen in the BiLSTM.

## 6.2 Recommended Further Work

The main limiting factor in deep learning projects is not seldom the data itself. The quality of the result can only be as good as the quality of the data provided. We believe that the data can be further refined than what was done in this project. There are strong reasons to suggest that a fully automatic classifier could compete with a trained speech therapist by eliminating the issues mentioned above. But for this to be achieved, we believe that the data gathering process must be standardized as much as possible to give high-quality, robust inputs to the network. The ideal audio recording would follow these guidelines:

1. Only the child's voice is present, or at least easily extracted by some structural knowledge about the recording. For example, if timestamps for each speaker would be made available, extracting the child's voice would be easy.
2. Based on the issues of labeling entire file contents, it would also be best if only the key utterances like "*Kicki kokar potatis*" and "*Titti tittar på TV*" are recorded. A high-quality recording would almost exclusively record key utterances that can point to which class the child should be assigned to.
3. To cement standardization, it would also be for the best if these utterances always came in the same order so that no file is different from the other based on speech content.

If these steps were realized, the network would have a much easier time finding and focus on the patterns that cause the severity of CLP speech. All audio frames would be of high quality and indicative of class level. Furthermore, the network would better be able to compare audio frames between audio files as the contents would be roughly similar given the now standardized recording. This would allow the network to easier find the correct differences between the frames compared to earlier when the speech content was different, which made CLP differences harder to find.

Examining research papers of similar work, most exclusively rely on short audio segments as their data. Good results were, for example, achieved by a study classifying hypernasality in CLP children and the underlying data consisted only of vowel utterances [40].

To further test this hypothesis of new data gathering methods, it could be worth taking the time to label the data used in this thesis. Manually extracting the same key utterances from each audio recording would be time-consuming but greatly standardize the data. After the refined data is obtained, attempting to train the same or a slightly altered version of the models should yield better results and thus further strengthen the case to change, or at least complement, the data gathering method.

# Bibliography

[1] S. Abhinav. *5 Techniques to Prevent Overfitting in Neural Networks.* 2019.

[2] H. Bredin et al. “pyAnnote.audio: neural building blocks for speaker diarization”. In: *ICASSP 2020, IEEE International Conference on Acoustics, Speech, and Signal Processing.* Barcelona, Spain, May 2020.

[3] J. Brownlee. *Deep Learning for Computer Vision.* 2019.

[4] J. Brownlee. *Long Short-Term Memory Networks with Python.* 2017.

[5] A. Cheveign and H. Kawahara. “YIN, a fundamental frequency estimator for speech and music.” In: *The Journal of the Acoustical Society of America* 111 4 (2002), pp. 1917–30.

[6] T. Dodderi et al. “Spectral Analysis of Hypernasality in Cleft Palate Children: A Pre-Post Surgery Comparison”. In: 10 ().

[7] K. Doshi. *Audio Deep Learning Made Simple (Part 2): Why Mel Spectrograms perform better.*

[8] V. Feng. *An Overview of ResNet and its Variants.* 2017.

[9] E. Flemming. “Linguistic Phonetics: The source-filter model of speech production”. Massachusetts Institute of Technology: MIT OpenCourseWare, 2015.

[10] T. Giannakopoulos and A. Pikrakis. *Introduction to Audio Analysis: A MATLAB Approach.* 1st. Academic Press, Inc., 2014.

[11] A. Graves and J Schmidhuber. “Framewise phoneme classification with bidirectional LSTM and other neural network architectures”. In: *Neural Networks* 18.5 (2005), pp. 602–610.

[12] A. Graves and J. Schmidhuber. “Framewise phoneme classification with bidirectional LSTM networks”. In: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* Vol. 4. 2005, 2047–2052 vol. 4.

[13] F. Hallberg and I. Nordin. “Talutveckling vid läpp-käk-gomspalt: samband mellan konsonantproduktion vid 18 månader och tre år”. 2019.

[14] S. Haşim, A. Senior, and F. Beaufays. “Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition”. In: (2014).

[15] S. Hershey et al. “CNN Architectures for Large-Scale Audio Classification”. In: *International Conference on Acoustics, Speech and Signal Processing (ICASSP).* 2017.

[16] S. Hochreiter and J. Schmidhuber. "Long Short-term Memory". In: *Neural computation* 9 (Dec. 1997), pp. 1735–80.

[17] D. Ishak. "A speaker recognition system based on vocal cords' vibrations". PhD thesis. Dec. 19, 2017.

[18] A. Jakobsson. *An introduction to Time Series Modeling.* Malmö: Studentlitteratur, 2019.

[19] C. Jo and J. Kim. "Segregation of voiced and unvoiced components from residual of speech signal". In: *Journal of Central South University* 19.2 (Feb. 1, 2012), pp. 496–503.

[20] E. Kang. *Long Short Term Memory (LSTM): Concept.* 2017.

[21] J. Kreiman and D. Sidtis. *Foundations of voice studies : an interdisciplinary approach to voice production and perception.* Wiley-Blackwell, 2011.

[22] S. Kulkarni. "Frequency Domain and Fourier Transforms". In: *Information Signals.* 2002.

[23] M. Lavechin et al. "End-to-end Domain-Adversarial Voice Activity Detection". In: (2020).

[24] Guo-She Lee et al. "Voice Low Tone to High Tone Ratio: A Potential Quantitative Index for Vowel[$ssra!$ :]and Its Nasalization". In: *IEEE transactions on bio-medical engineering* 53 (July 2006), pp. 1437–9.

[25] V. Mathad et al. "Deep Learning Based Prediction of Hypernasality for Clinical Applications". In: May 2020, pp. 6554–6558.

[26] P. McLeod and G. Wyvill. "A smarter way to find pitch". In: Jan. 2005.

[27] PerceptiLabs. *Five Common Hyperparameters.* 2020.

[28] Ken C. Pohlmann. *Principles of digital audio.* McGraw-Hill, 2000.

[29] L. Rabiner and R. Schafer. *Introduction to Digital Speech Processing.* 2007.

[30] B. Ramsundar and R. Bosagh Zadeh. *TensorFlow for Deep Learning.* O'Reilly Media, Inc., 2018.

[31] M. Ravanelli and Y. Bengio. "Speaker Recognition from Raw Waveform with SincNet". In: *2018 IEEE Spoken Language Technology Workshop (SLT)* (2018), pp. 1021–1028.

[32] A. Richmond. *GoogLeNet Explained.* 2020.

[33] M.D Sahidullah and Saha Goutam. "Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition". In: *Speech Communication* 54.4 (2012), pp. 543–565.

[34] P. Skalski. *Gentle Dive into Math Behind Convolutional Neural Networks.* 2019.

[35] S. Stevens and J. Volkmann. "The Relation of Pitch to Frequency: A Revised Scale". In: *The American Journal of Psychology* 53.3 (1940), pp. 329–353.

[36] I. Titze. *Principles of Voice Production.* National Center for Voice and Speech, 2000.

[37] Q. Wang. *Awesome Diarization.* 2021.

[38] Q. Wang et al. "Speaker diarization with lstm". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE. 2018, pp. 5239–5243.

[39] X. Wang, Y. Zhao, and F Pourpanah. "Recent advances in deep learning". In: *International Journal of Machine Learning and Cybernetics* 11.4 (Apr. 1, 2020), pp. 747–750.

[40] X. Wang et al. "HypernasalityNet: Deep recurrent neural network for automatic hypernasality detection". In: *International Journal of Medical Informatics* 129 (2019), pp. 1–12.

[41] W. Whitney. *A diagram of a convolutional layer and a pooling layer applied to an image.* Licensed under creative commons 4.0. 2014.

[42] J. Wolfe. *Voice Acoustics: an introduction to the science of speech and singing.* University of New South Wales. (Visited on 06/10/2021).

[43] R. Yin, H. Bredin, and C. Barras. "Speaker Change Detection in Broadcast TV Using Bidirectional Long Short-Term Memory Networks". In: *INTERSPEECH.* 2017.

[44] J. Zhang et al. "Automatic hypernasality grade assessment in cleft palate speech based on the spectral envelope method". In: *Biomedical Engineering / Biomedizinische Technik* 65.1 (2020), pp. 73–86.

[45] Zsiga, E. *The Sounds of Language: An Introduction to Phonetics and Phonology (Linguistics in the World).* Wiley-Blackwell.

[46] C3SE. *About Alvis.* Link. 2021.

[47] Electrical Engineering Stack Exchange. *In digital systems do we discretize both time and magnitude or only time?* Link.

[48] Mathworks, Inc. *Harmonic Ratio.* Link.

[49] Mathworks, Inc. *Spectral Descriptors.* Link.

[50] Mathworks, Inc. *Speech Emotion Recognition.* Link.

[51] Understanding FFTs and Windowing. *National Instruments.* 2019.

[52] Wikimedia Commons, the free media repository. Cleftpalate1.png. Link. 2020.

[53] Wikimedia Commons, the free media repository. Cleftpalate2.png. Link. 2020.

[54] Wikimedia Commons, the free media repository. Cleftpalate3.png. Link. 2020.

[55] Wikimedia Commons, the free media repository. Multi-Layer Neural Network-Vector-Blank.svg. Link. 2020.

# A

# Appendix 1

| Layer | Type | Description |
|---|---|---|
| 1 | Input | Input layer of 33x20 |
| 2 | BiLSTM | Hidden units 128, sequence output |
| 3 | BiLSTM | Hidden units 128, sequence output |
| 4 | BiLSTM | Hidden units 128, last output |
| 5 | FC | 3 output neuron fully connected layer |
| 6 | Output | Softmax with class conditional probabilities |

**Table A.1:** BiLSTM network architecture

| Layer | Type | Description |
|---|---|---|
| 1 | Input | Image input layer of 96x64 |
| 2 | Conv | 12x3x3 filters with stride 2 |
| 3 | Batch | Batch normalization layer |
| 4 | Relu | Relu layer |
| 5 | Pool | Max pool Layer 3x3 filter, stride 2 |
| 6 | Conv | 24x3x3 filters with stride 2 |
| 7 | Batch | Batch normalization layer |
| 8 | Relu | Relu layer |
| 9 | Pool | Max pool Layer 3x3 filter, stride 2 |
| 10 | Conv | 48x3x3 filters with stride 1 |
| 11 | Batch | Batch normalization layer |
| 12 | Relu | Relu layer |
| 13 | Conv | 48x3x3 filters with stride 1 |
| 14 | Batch | Batch normalization layer |
| 15 | Relu | Relu layer |
| 16 | Pool | Max pool Layer 3x3 filter, stride 1 |
| 17 | Drop | Dropout layer with 0.2 probability |
| 18 | FC | 3 output neuron fully connected layer |
| 19 | Output | Softmax with 3 class conditional probabilities |

**Table A.2:** CNN network architecture

| Layer | Type | Description |
|---|---|---|
| 1 | VGGish | Complete VGGish architecture, excluding the final layer |
| 2 | FC | 3 output neuron fully connected layer |
| 3 | Output | Softmax classification layer |

**Table A.3:** T.L VGGish network architecture

| Parameter | Setting | Description |
|---|---|---|
| Mini batch size | 64 | Samples seen between weight updates |
| Initial learning rate | 0.001 | Initial step size for optimizer |
| Optimizer | Adam | Minimization algorithm |
| Loss function | Cross-entropy | A measure to quantify errors |
| Decay | 0.1 | Learning rate factor decay |
| Epochs | 10 | How many times the network sees all data |

**Table A.4:** BiLSTM training settings

| Parameter | Setting | Description |
|---|---|---|
| Mini batch size | 16 | Samples seen between weight updates |
| Initial learning rate | 0.001 | Initial step size for optimizer |
| Optimizer | Adam | Minimization algorithm |
| Loss function | Cross-entropy | A measure to quantify errors |
| Decay | 1 | Learning rate factor decay |
| Epochs | 2 | How many times the network sees all data |

**Table A.5:** CNN training settings

| Parameter | Setting | Description |
|---|---|---|
| Mini batch size | 512 | Samples seen between weight updates |
| Initial learning rate | 3e-04 | Initial step size for optimizer |
| Optimizer | Adam | Minimization algorithm |
| Loss function | Cross-entropy | A measure to quantify errors |
| Decay | 0.1 | Learning rate factor decay |
| Epochs | 25 | How many times the network sees all data |

**Table A.6:** T.L VGGish training settings

Master's Theses in Mathematical Sciences 2021:E69
ISSN 1404-6342

LUTFMS-3434-2021

Mathematical Statistics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lu.se/