FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of master's thesis

Title:	Lip Reading
Student:	Bc. Justína Kušpálová
Supervisor:	Mgr. Martin Mareš
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2024/2025

Instructions

Automatic text transcription from videos has received a lot of attention in recent years. This thesis focuses on visual speech recognition, as other similar works usually use both video and audio signals. Visual information is not affected by acoustic noise and can be used when sound is missing or corrupted. The goal is to explore the performance of current lip reading deep-learning models on Slovak or Czech videos since no publicly available studies have been conducted on lip reading in these languages.

1) Survey current state-of-the-art deep learning methods for lip reading.
2) Explore and choose the correct existing dataset for the given task.
3) Based on the research, implement at least 2 models suitable for sentence-level visual speech recognition.
4) Create an experimental test dataset in Slovak or Czech language.
5) Evaluate the performance of implemented models on datasets.
6) Discuss the limitations of implemented methods on the Slovak/Czech dataset, and research possible improvements.

Resources:
J. S. Chung et al., "Lip reading sentences in the wild," in CVPR, 2017, pp. 3444–3453.
P. Ma et al., "Visual Speech Recognition for Multiple Languages in the Wild," Nature Machine Intelligence, pp. 930–939, 2022.

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 6 October 2023 in Prague.

Master's thesis

LIP READING

Bc. Justína Kušpálová

Faculty of Information Technology
Department of Applied Mathematics
Supervisor: Mgr. Martin Mareš
May 9, 2024

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis: Kušpálová Justína. *Lip reading.* Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2024.

Contents

List of Figures

List of Tables

List of code listings

I would like to thank my supervisor Mgr. Martin Mareš and his colleague Mgr. Adam Szabó for their guidance, advice and patient approach. I also wish to thank my family and friends who supported me throughout my studies. Lastly, I am thankful to my boyfriend whose encouragement and understanding were crucial during the writing of this thesis.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis. I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

In Prague on May 9, 2024

Abstract

This diploma thesis focuses on the automatic lip reading for Czech language. This automatic speech recognition is performed in uncontrolled environments that are characterized by varied lighting conditions, diverse backgrounds and multiple different speakers all of which complicate the visual processing of lip movements. This thesis inspects existing approaches for other languages and applies and evaluates similar principles to the Czech language using deep learning. Recognizing the limited resources in non-English datasets for lip reading, this work also involves the creation of a tailored train, validation and test dataset to facilitate model training and evaluation. The evaluation is done using newly created Czech lipreading testing dataset created with help of Czech Television as a part of this thesis.

Keywords automatic visual speech recognition, lip reading in uncontrolled environment, transformer architecture

Abstrakt

Táto diplomová práca sa zaoberá automatickým čítaním pier v českom jazyku. Rozpoznávanie reči prebieha v nekontrolovanom prostredí, pre ktoré sú charakteristické rôzne svetelné podmienky, rôznorodé pozadie a rôzni rečníci. Všetky spomínané faktory komplikujú vizuálne spracovanie pohybov pier. Na základe existujúcich riešení pre iné jazyky, táto práca implementuje a vyhodnocuje podobné princípy v českom jazyku. Z dôvodu limitovanch zdrojov dát pre iné jazyky ako je angličtina, táto práca taktiež obsahuje proces vytvárania trénovacieho, validačného a testovacieho datasetu v českom jazyku. Vyhodnotenie následne prebieha na novovytvorenom datasete, ktorý vznikol v spolupráci s Českou Televíziou.

Klíčová slova automatické rozpoznávanie reči, čítanie z pier v nekontrolovanom prostredí, transformer architektúra

Seznam zkratek

VSR	Visual Speech Recognition
AVSR	Audio-visual Speech Recognition
ROI	Region of Interest
CNN	Convolutional Neural Network
CTC	Connectionist Temporal Classification
ReLU	Rectified Linear Unit
LRS2	Lip reading sentences 2
LRS3	Lip reading sentences 3
Conformer	Convolution-augmented transformer
MHSA	Multi-headed self-attention
EOS	End of Sequence

Introduction

Lip reading is a technique of understanding speech by observing the movements of the lips and facial expressions without hearing the accompanying sounds. During this process, other visual clues, such as body language, eye movements, and hand gestures, can supplement lip reading and enhance comprehension. Automatic lip reading entails the process of decoding spoken language from videos by analyzing the movements of a speaker's lips using various techniques. It is an interdisciplinary subject that involves knowledge from different fields, such as image processing, computer vision and natural language processing.

Lip reading has a wide application in different situations. Visual information is the only source of information for people with hearing impairments if there are no assistive sign language or subtitles in a video. Visual speech recognition in security supports surveillance capabilities, enabling the analysis of silent video footage. Some other avenues include understanding a person speaking in a noisy environment, transcribing and dubbing archival silent films or some less ethical, such as off-microphone exchanges between politicians or celebrities.

This thesis focuses on the automatic text transcription of video content. Transcription is handled using visual input without an audio signal. I explore current deep-learning models for lip reading and implement them to read lips from videos in the Czech language. This visual speech recognition is performed at the sentence level and operates within unconfined, real-world environments.

An integral part of creating such a model is the preparation of a suitable dataset. Since no publicly available studies have been conducted on lip reading in the Czech language, I had to create an experimental dataset. I explored multiple options, but some limitations have to be considered regarding the criteria such a dataset must meet. These conditions include high-resolution video footage, a diverse range of speakers, backgrounds, and language variety. This led me to choose data collected from Czech Television shows that meet all these criteria and are free for academic use. Selecting this approach forced me to deal with legal issues concerning the sharing permissions of the acquired dataset. This dataset and its collection hold value for future endeavours in Czech automatic speech recognition, addressing the prevailing challenge of data scarcity in this field.

The goals of this thesis are:

- survey state-of-the-art deep learning methods for lip reading
- design, implement and evaluate a lip reading models suitable for the Czech language
- create an experimental Czech dataset, evaluate an method for creating such datasets

- evaluate the performance of implemented models
- experiment with possible improvements.

In Chapter 1, I will introduce the visual perception of speech and define basic linguistic terms. Chapter 2 provides a theoretical background for the elements used in this work. Chapter 3 presents a general framework for automatic visual speech recognition (VSR) and reviews the state-of-the-art methods used in this domain. Chapter 4 provides an exploration of existing datasets for lip reading and describes the process of creating a custom Czech dataset inspired by the datasets surveyed. It explains the steps involved in dataset creation, from the initial design and collection of video samples to the annotation process. Chapter 5 describes the background of the methods and architectures of deep learning models that I adopted. Chapter 6 shows the progressive development of selected models using the Czech data. The last chapter evaluates the results and experiments with possible improvements to optimize the performance.

Chapter 1

Speech and linguistics

People's speech patterns vary widely. These patterns are influenced by geographical, social, and personal factors. Some individuals clearly articulate, forming distinct shapes with their lips. That makes lip reading easy. On the contrary, some people speak rapidly or mumble, causing their lip movements to be more challenging to interpret. Some speakers are naturally more expressive and convey emotions and emphasis through exaggerated lip movements, while others exhibit minimal facial movements and appear almost visually "speechless". In addition, accents play a significant role in how speech appears. Different accents can influence the shape and movement of the lips. All of these factors lead to a wide range of visual clues.

1.1 Perception of speech

Everyday speech is usually too fast to be lipread easily. The typical speaking involves 13 sound units per second, yet only 8–10 lip movements per second may be consciously seen by the eye. Therefore, our ability to comprehend all visual information during speech is limited [1].

Many speech patterns are similar, and some lip movements look similar, but they sound different and have different meanings. This unambiguity is a significant obstacle in visual speech recognition, leading to confusion and doubt. Even people with hearing impairment who use lip reading as a support tool to enhance understanding achieved an accuracy of only $17 \pm 12\%$ for a limited subset of monosyllabic words and $21 \pm 11\%$ for spondaic words[1] [2]. It illustrates the complexity of lip reading, even for professionals.

Although speech perception is considered an auditory skill and lip reading is used most extensively by people with severe hearing impairment, even people without any hearing disabilities process some visual stimuli to support auditory comprehension, making it intrinsically multimodal. This was demonstrated in the McGurk effect [3], which describes an auditory-visual illusion. The best-known case of this effect refers to a voice that says /*ba*/ on a face that articulates /*ga*/. That would result in perceiving /*da*/ on the receiver side. It indicates that the movements of the mouth that we see can influence what we believe to be hearing. Even when our sensory organs function correctly, errors can still occur when interpreting signals.

1.2 Linguistics background

This section is describing linguistic background [4].

[1] compound

Phone is any distinct speech sound which is not specific to any language, regardless of whether the sound is critical to the meaning of words. The standard notation for a phone is [ ].

Phoneme is a language's smallest detectable unit of sound. Unlike a phone, it serves to distinguish words from each other. The standard notation for a phoneme is //.

A pair of words that differ in one phoneme is called a **minimal pair**.

If two phones sound similar and do not change the meaning of the word, they are **allophones** of the same phoneme. For example, in English, $[k]$ and $[k^h]$ are allophones of a phoneme $/k/$. If the speaker uses one instead of the other, the word's meaning will not change; even though sounding odd, it will still be recognized. In some languages, these two phones are considered significantly different sounds, and substituting one for the other changes the word's meaning. Therefore, these languages have two separate phonemes $k/^h/$ and $/k/$.

The exact number of phonemes can vary slightly depending on accent and dialect, but generally, there are 44 phonemes in English [5] and 39 phonemes in the Czech language [6].

While phoneme is the basic unit of sound, a **viseme** is a visually distinctive unit. [7, 8]. It represents the position of the face and mouth during the speech. The ambiguity problem in lip reading is that several phonemes are assigned to one viseme, making it sometimes difficult to differentiate without additional information [9]. The differences in some phonemes are produced inside the mouth or throat, making it impossible to see and leading to this many-to-one phoneme-to-viseme mapping. Moreover, some phonemes can also alter or even vanish in appearance. An example is velar consonants, such as $/k/$ or $/g/$, which alter the position of the tongue in the palate based on the phoneme before or after them [10].

A set of words that sound different but look identical during speaking is called **homophenes**. They are a significant source of mis-lip reading.

There is no consensus among researchers on using a standard viseme table. That nasal and non-nasal pairs such as $[m]$ and $[b]$ or voiced and unvoiced pairs such as $[p]$ and $[b]$, $[k]$ and $[g]$, $[t]$ and $[d]$, $[f]$ and $[v]$, and $[s]$ and $[z]$ look identical during speech on the face.

There are about three times more phonemes than visemes in English. For instance, phonemes $/t, d, n, l/$ are represented by one viseme $/t/$ or phonemes $/p, b, m/$ are mapped to the viseme $/p/$. Thus, words such as park, bark, and mark are difficult for lip-readers to distinguish, as all look alike. Sometimes, the whole sentence might be confusing, such as the phrase "elephant juice", which lip-readers can interchange with the phrase "I love you" since they appear visually identical.

The statistical distribution of phonemes in a language is not even. Some words form phonemically similar clusters; on the other hand, there are words whose phonemic distribution is quite unique within a given language. The second group can be lipread without ambiguity since only one option exists for a word to fit.

Chapter 2
Theoretical background

2.1 Attention

People have always tried to teach computers how to process information in a manner similar to that of the human brain. The same applies to the attention mechanism, which attempts to imitate cognitive attention when a person selectively focuses on one or a small number of relevant things while disregarding others. The first idea of attention [11] was improved and described as "mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors" [12]. These three vectors are derived from the input through learned linear transformations. The weight of each value is then calculated using the compatibility function of the query with the corresponding key, and the output is computed as a weighted sum of these values.

Following the notation [12], considering Q as a vector of queries, K as a vector of keys, both of dimension d_k, V with dimension d_v as a vector of values and $\frac{1}{d_k}$ as a scaling factor for the output vector, Scaled Dot-Product Attention can be computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

This type of attention can be computed much faster than additive attention using optimized matrix multiplication. The scaling factor $\frac{1}{\sqrt{d_k}}$ is crucial to stabilize the gradients during learning, as it prevents the dot products from growing too large in magnitude, which can lead to vanishing gradients during backpropagation. Self-attention is a mechanism that adds context-related information to an input embedding, thus improving its information richness. It allows the model to assess the significance of various input sequence elements and dynamically modify their impact on the output.

2.1.1 Multi-Head Attention

In multi-head attention, these vectors Q, K and V are not used directly. Instead, each set is linearly projected h times with different linear projections. This results in h different sets of queries, keys, and values, where h is the parameter called number of heads.

The attention function is applied in parallel for each projected set of queries, keys, and values. Each head independently attends to information at different positions across the input sequence. It allows the model to better integrate information from different representational spaces. The operations within each head are independent of the others, making them highly parallelizable.

After each head has produced its output, these outputs are concatenated into a single matrix. This concatenated matrix is then linearly projected one more time to produce the final output of the multi-head attention layer. This projection integrates information from all the heads, allowing the model to leverage different representations from different heads.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_\text{h})W^O \tag{2.2}$$

where

$$\text{head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.3}$$

where the projection are parameter matrices $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$.

2.2 Transformer

RNN and similar sequence neural networks, such as LSTMs or GRUs, have had several issues. One of the most significant ones is slow computation for long sequences. It comes from their sequential nature, where for each sequence of length n, n steps are required to map the output. Their computational power is insufficient for complex tasks like reading the lips. That is why the transformer-based model is more suitable for this task.

A standard transformer consists of an encoder that processes the input data and a decoder that generates the output. A general transformer architecture can be seen in Figure 2.1.

2.2.1 Encoder

An encoder takes positionally encoded input embedding. The encoder module is made up of a multi-head self-attention mechanism followed by a position-wise fully-connected feed-forward neural network. A residual connection exists around both sub-layers, followed by layer normalization. The residual connections and layer normalization help stabilise the learning and prevent the vanishing gradient problem. This module is repeated multiple times, depending on the architecture.

2.2.2 Decoder

The decoder is also composed of several identical layers stacked on each other. It takes a positionally encoded output that is shifted to the right by the start of the sequence $< sos >$ token. The output embedding is given to a masked multi-headed self-attention mechanism. It allows the decoder to focus on different parts of the output sequence it has generated so far, which is crucial for generating coherent and contextually appropriate text. Unlike in the encoder, where self-attention layers look at the entire input sequence, self-attention in the decoder is masked to prevent future positions from being accessed. This masking ensures that the prediction of ith position depends only on the known outputs at positions up to i. All undesired values – illegal connections – are set to $-\infty$ before applying softmax. This will result in assigning a value close to 0 after softmax and thus not considering them.

Then, a layer of cross multi-head attention over encoder output follows. The features of the previous self-attention block are used as Q, and the representations K and V come from the encoder output. This mechanism allows the decoder to focus on relevant parts of the input sequence using the encoder's outputs to guide the generation of the output sequence. The last sub-layer is the feed-forward network. Similarly, residual connection and layer normalization remain the same as in the encoder. The final layer of the decoder outputs a vector of scores,

typically passed through a softmax layer to form a probability distribution over possible output tokens.

During the training phase, the decoder uses a training strategy called "teacher forcing". Instead of using the model's own predictions from the previous time step as inputs, it uses actual target outputs since the target outputs are known. During the inference phase, the model generates an output at each step based on its previous outputs. This approach can accelerate convergence.

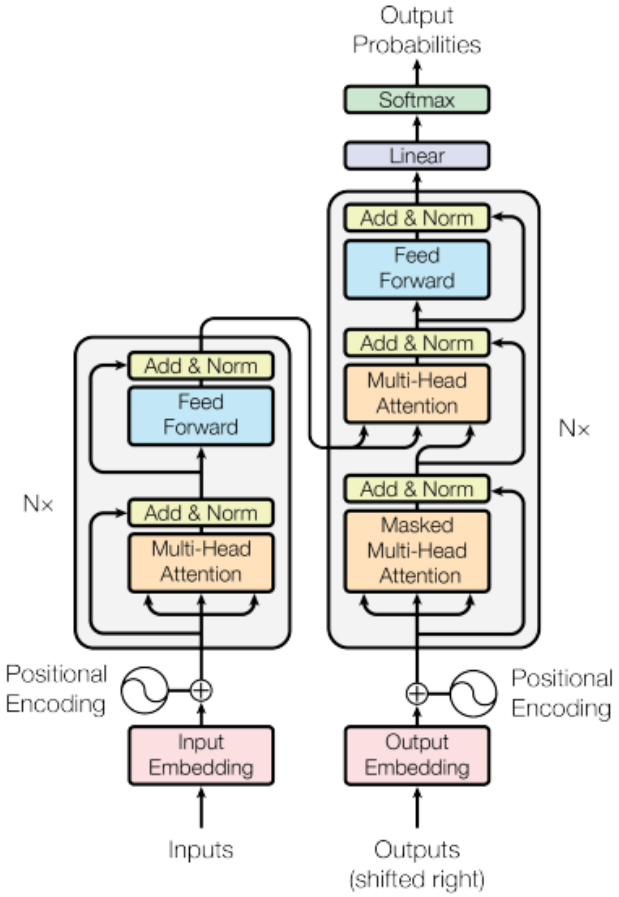

Figure 2.1 Transformer architecture [12].

2.2.3 Positional encoding

Since the Transformer does not inherently process data in sequence, positional encodings are added to preserve sequential information. One of the many examples involves using sine and cosine functions of different frequencies [12]:

$$\begin{aligned} PE_{pos,2i} &= \sin\left(pos/10000^{2i/d_{model}}\right) \\ PE_{pos,2i+1} &= \cos\left(pos/10000^{2i/d_{model}}\right) \end{aligned} \tag{2.4}$$

where pos is the position and i is the dimension.

2.3 Conformer

The convolutional-augmented transformer combines the strengths of CNNs and Transformers. CNNs successfully capture local context but need a huge amount of parameters to capture global information, whereas transformer models are good at capturing content-based global interactions while suffering from an inability to extract fine-grained local feature patterns. It has been shown [13] that the combination of self-attention and convolution increases performance compared to using them separately.

Conformer encoder is composed of convolution subsampling, linear layer, dropout and then a set of stacked Conformer blocks. Each conformer block is made up of four modules: feed-forward, self-attention, convolution and again feed-forward module. The Conformer employs Macaron-style [14] feed-forward networks, where feed-forward layers are placed before and after the self-attention and convolution modules. The architecture of the Conformer encoder is shown in Figure 2.2.
Formally, the output y_i of i-th conformer block for input x_i is:

$$\begin{aligned} x'_i &= x_i + \frac{1}{2}\mathrm{FF}(x_i) \\ x''_i &= x'_i + \mathrm{MHSA}(x'_i) \\ x'''_i &= x''_i + \mathrm{Conv}(x''_i) \\ y'_i &= \mathrm{Layernorm}(x'''_i + \frac{1}{2}\mathrm{FF}(x'''_i)) \end{aligned} \tag{2.5}$$

2.4 3D CNN

Video-based data can benefit from 3D convolutions. As the name suggests, the 3-dimensional kernel can slide in 3 dimensions, making it possible to learn both spatial and temporal features and thus analyze the relationship between frames in time [16, 17]. On the other hand, this type of network tends to lose the ability to extract fine-grained feature information.

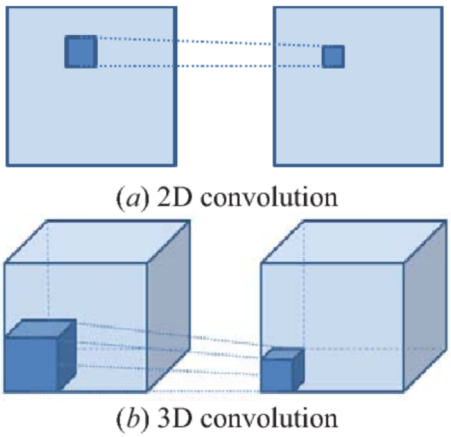

Figure 2.3 Difference between (a) 2D and (b) 3D CNN [18].

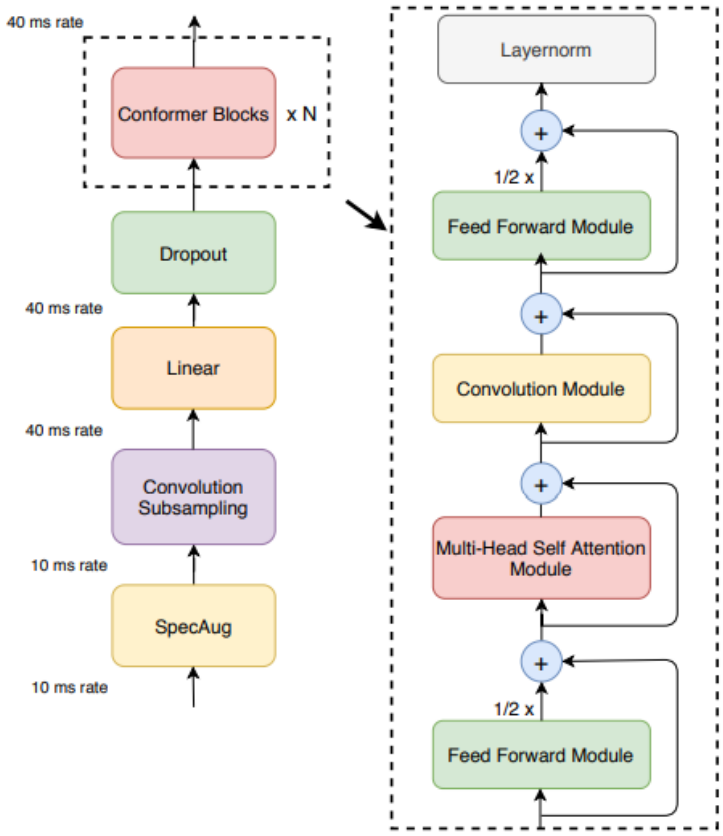

Figure 2.2 Architecture of the Conformer encoder [15]. Conformer comprises of two macaron-like feed-forward layers with halfstep residual connections sandwiching the multi-headed self-attention and convolution modules. This is followed by a layer normalization.

2.5 ResNet

The deeper a convolutional neural network is, the greater the ability to capture any space due to the many learnable weights it has. However, this works only to some depth, and then the performance degrades [19]. This is caused by vanishing gradients and degradation of training accuracy. ResNets address both of these problems. They use skip connections where gradients can flow directly backwards from later layers, which prevents them from shrinking to zero after several applications of the chain rule in backpropagation. There are various architectures of ResNets; the model I adopted uses ResNet-18. The architecture can be found in the Appendix Section A.1 in Figure A.1 and Figure A.2 displays the ResNet-18 diagram.

Chapter 3
Related work

A typical automatic lip reading system [20, 21], takes visual input at first. It is usually a video that is sampled into image frames and these frames are pre-processed. There are different approaches to pre-processing; however, they all keep the idea of extracting the region of interest (ROI). Relevant features are selected and further mapped in a lower-dimensional representation. This happens in the so-called front-end layer. The back-end layer maps these lower-dimensional facial movement feature vectors into encoded speech parts. These are decoded into classes or units to output characters, words, or sentences.

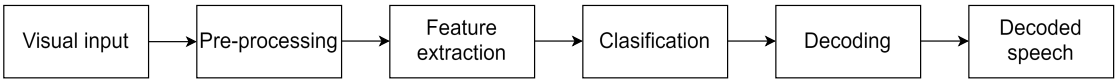

Figure 3.1 Typical framework of a lipreading model.

Figure 3.1 is an overview of a general automatic lip reading model for basic understanding; individual parts will be further described and surveyed in this chapter.

3.1 Deep learning Visual Speech Recognition

The first deep learning VSR systems [22, 23, 24, 25] combined methods like Hidden Markov models (HMM) [22, 23, 24] or Support Vector Machine [25] blocks with deep neural networks. In these models, neural networks were only used as feature extractors. When neural networks replaced the entire processing chain [26], the performance improved significantly. Today, the use of neural networks represents the most effective method for lip reading, achieving significantly higher accuracy [21, 27] in automatic visual speech recognition than other approaches.

3.1.1 Pre-processing

The first step in pre-processing a recorded video of someone speaking is to sample the recording into individual image frames. The region of interest, specifically the lips, is identified and isolated from the raw image data. This process entails detecting the face, pinpointing the lips, and extracting the lip area from the video frames. Modifications such as cropping are applied to reduce the computational complexity during the training procedure.

3.1.2 Feature extraction

The purpose of the feature extraction or, so-called, front-end layer is to separate essential features from non-essential features and then transform these high-dimensional features into low-dimensional space.

2D CNNs have been widely used for feature extraction [28, 29, 30]. They can extract the spatial features of each image. 3D CNN can extract both time and space information from continuous frames and capture the relationship between them. Some authors used them [31, 32] for feature extraction.

As a next step, approaches that combine 2D and 3D CNNs [33] were developed. First, a 3D convolution is applied to process continuous information, and then deep 2D CNN follows, extracting local features. Methods based on this 2D and 3D CNN combination have become very popular for feature extraction in the lip reading domain [34, 35, 21, 27]. This architecture can be seen in Figure 3.2.

3.1.3 Classification

The previous phase – the feature extraction – is generally well-developed because similar methodologies are employed in other areas of deep learning, such as image and video analysis. However, the subsequent phase in the lip reading process, the speech classification, still poses significant challenges. After features are extracted, the correct classification of these features into spoken words or phrases remains complex.

A classification or, so-called, back-end layer consisting only of a softmax layer was enough for classification from the predefined vocabulary [36]. However, this is impossible in an unconstrained environment where individuals say longer words or sentences. For this reason, many visual speech recognition systems include more complex back-ends with neural networks to analyze temporal sequences. This includes RNN-based architectures such as LSTM [37], GRU [38] or their bidirectional versions Bi-LSTM [39] and Bi-GRU [40].

In recent studies [41, 42], Transformer architectures have introduced in lipreading models. They allow parallel computation and are better at capturing long-term dependencies, as well as in terms of their overall performance on lipreading tasks.

3.1.4 Decoding

Early VSR systems worked with simple recognition tasks like character or digit recognition. Then, models that tried to predict a word from a constrained vocabulary of words or phrases, usually in a controlled environment, emerged. These systems have evolved into more intricate and authentic settings, focusing on continuous lip reading at the sentence level.

There are various ways to interpret and classify lip movements. It could be phonemes [43, 44], visemes [45, 46], ASCII characters [39, 47], words [48, 49, 37] or sentences. Each of these techniques has its benefits and disadvantages as well. The diagram showing the interpretation of lip movements can be seen in Figure 3.3.

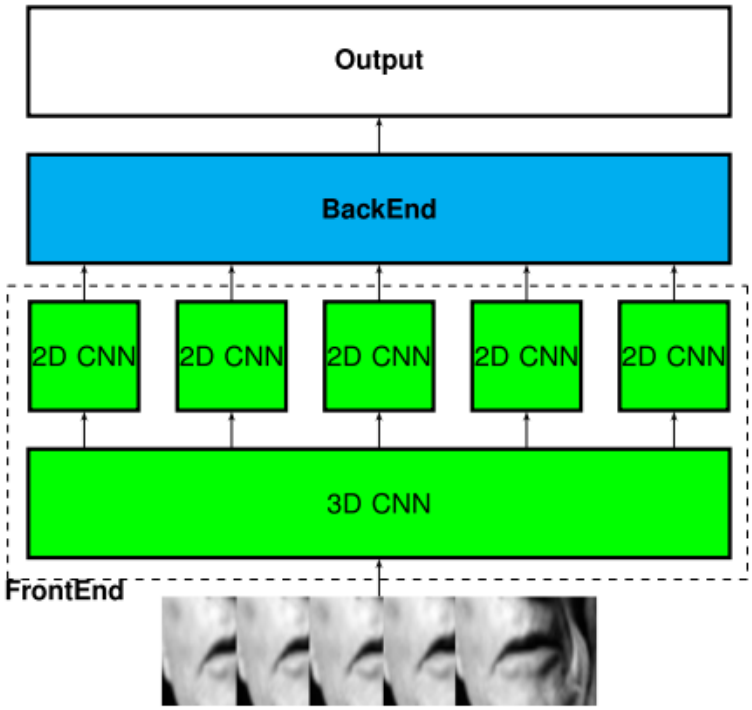

Figure 3.2 Front-end composed of 3D and 2D CNN [20].

3.1.4.1 Visemes

There are fewer classes to predict when using the visemes since there are fewer visemes than ASCII characters or words. This reduces the computational bottleneck. Visemes do not need previously learned lexicons, which means that words not observed during training could be predicted correctly. Many different languages share similar visemes, offering the possibility of generalizing the model and decoding speech from various languages. However, their performance is less satisfactory because of the ambiguity during the decoding procedure. This process involves two steps; movement-to-viseme conversion and then viseme-to-word conversion. The second one mentioned is a source of incorrectly predicted words since one set of visemes can be mapped to multiple possible words.

3.1.4.2 Phonemes

Unlike classification using visemes, classification using phonemes faces the exact opposite problem. There is less ambiguity involved during the conversion of phonemes to words. This is because there are naturally fewer words that sound the same than words that appear the same visually. On the other hand, many phonemes look similar, which is a source of errors in decoding movements to phonemes in the first place.

3.1.4.3 Words and phrases

Lip-reading systems with words or even phrases as classes emerged when more audiovisual training data were available, covering a larger vocabulary set. Having words as classes for prediction means having the same number of classes as the number of unique words in the vocabulary. Some of the largest and most current lip-reading corpuses cover thousands of distinct words. It

is computationally expensive and the model is not able to generalize beyond what it has seen during the training phase.

3.1.4.4 ASCII characters

The most recent methods for automated lip reading focus mainly on interpreting continuous speech segments with ASCII characters that serve as the classes to be recognized. Conditional dependence relationships between ASCII letters make it possible to model a natural language, making it easier to predict characters or words. However, this is still a difficult issue because phrases can cover a large vocabulary and contain utterances that might not have been present during the training phase. The character combinations that have been seen as patterns during training are the basis for word prediction.

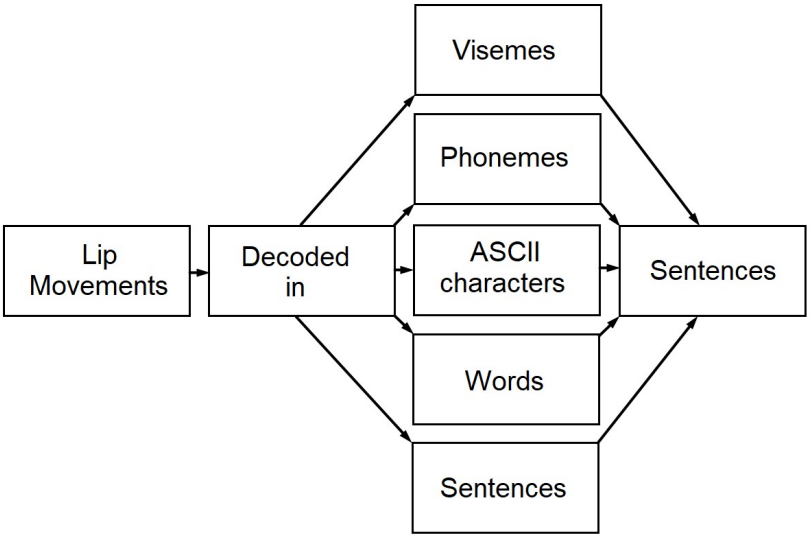

Figure 3.3 Interpretation of lip movements.

Chapter 4
Dataset

One of the major obstacles in training a lip reading model is the lack of data. A robust model that is able to generalize beyond the limited domain and controlled environment requires the dataset to be rich in vocabulary and to contain different poses and angles of the speaker, as well as various illumination levels. Creating such data that needs to be labelled takes an enormous amount of time. In this section, I will describe some of the most familiar datasets in this domain that were the source of inspiration for creating my experimental dataset.

LRS2 [47], also known as Lip Reading Sentences 2, is a collection of thousands of spoken sentences from various individuals captured in a "in the wild" setting. The videos include naturalistic speech variations, different accents and varying light conditions. This complexity makes it an excellent resource for developing more sophisticated and adaptable VSR systems. Models trained on the LRS2 dataset have shown remarkable improvements in word error rates (WER) compared to earlier benchmarks [27, 50].

This dataset consists of 144,482 video clips with a total duration of 224.1 hours. Videos are collected from BBC programs; each sentence is up to 100 characters long. It is one of the largest existing resources for visual speech recognition.

A sample taken from this dataset is displayed in Figure 4.1. The dataset statistics can be found in Table 4.1.

Set	# utterances	# word instances	Vocab	Duration in hours
Pre-train	96,318	2,064,118	41,427	195
Train	45,839	329,180	17,660	28
Validation	1,082	7,866	1,984	0.6
Test	1,243	6,663	1,698	0.5

Table 4.1 LRS2 dataset statistics.

Figure 4.1 A sample from LRS2.

The pre-train data contain video and transcript with timestamps for each word, for example:

```
Text:  AND THEN THEY FOLD OUT AND THEY BECOME BONE AND FUSE

AND 0.10 0.19
THEN 0.19 0.33
THEY 0.33 0.50
FOLD 0.65 1.02
OUT 1.02 1.74
AND 1.74 1.90
THEY 1.90 2.02
BECOME 2.02 2.52
BONE 2.52 2.99
AND 2.99 3.12
FUSE 3.12 3.61
```

Train, validation, and test sets contain transcript without timestamps:

```
Text:  IT OCCURRED IN THE MIDDLE OF NOWHERE
```

LRS3 [51], also known as Lip Reading Sentences 3, is an extension of its predecessor, LRS2. It is primarily derived from TED Talks videos, which provide a diverse range of speakers and speech content in a somewhat controlled environment. The dataset contains 151,819 utterances (438.9 hours), aiming to cover a broad spectrum of phonetic contexts. It is also a significant resource for lip reading, and I experimented with models pre-trained on this dataset as well. A sample taken from this dataset is displayed in Figure 4.2. The dataset statistics can be found in Table 4.1.

Set	# utterances	# word instances	Vocab	Duration in hours
Pre-train	118,516	3.9M	51k	408
Trainval	31,982	358k	17k	30
Test	1,321	10k	2k	0.9

Table 4.2 LRS3 dataset statistics.

The structure of the pre-train, train-validation and test set remains the same as in LRS2.

Figure 4.2 A sample from LRS3.

UWB-07-ICAVR [52] is a Czech audio-visual database. This corpus is made up of 10,000 continuous utterances from 50 different speakers. Each speaker said 200 sentences, the first 50 of which were the same for everyone and the remaining 150 of which were unique. Each sentence is recorded six times under different illumination conditions. This dataset could have been a good starting point to build upon; however, it is not publicly available.

4.1 Czech dataset

Training a Czech lip reading model is not possible without Czech training data. Since there is no publicly available dataset in the Czech language, I had to create a training data set. Creating a dataset for training a deep learning lip reading model presents several complex challenges. Initially, I had to identify and gather suitable sources of data. Once these sources are collected, the next task is to detect sequences within the videos that contain speech. Then, these sequences need to be extracted and segmented. Ensuring that each video segment features a face with visible lip movements is crucial. Finally, accurate transcription of the spoken words in these sequences is required. This procedure is critical for developing a robust lip reading model capable of performing well in real-world scenarios.

4.1.1 Selecting the resources

The first idea was to collect videos from the Chamber of Deputies, Parliament of the Czech Republic. The data are provided free of charge; the use of the data is conditioned by indicating the source of the data and possibly the date of data processing.

However, this dataset faces significant limitations. The environment of this data source is relatively controlled – all videos are from the same environment, recorded from the same distance and angle. There is a lack of diversity in the data – parliamentary speeches tend to have high repetition and redundancy. Speakers of parliament frequently reiterate vital points to ensure clarity. As a result, the language used can be uniform and limited in scope, focusing predominantly on specific topics. This repetition diminishes the linguistic variety and the range of expressions captured in the dataset, which can significantly impact the robustness and generalizability of models trained on these data. Consequently, datasets derived from parliamentary proceedings may not adequately reflect the rich, diverse linguistic nuances found in more varied language sources. Additionally, the limited number of speakers in parliamentary settings further restricts the diversity of the dataset, not just in terms of language but also in visual variety.

Datasets derived from parliamentary speeches also face challenges related to the physical setup of the speaking environment. Speakers often look down at their notes while delivering speeches or have their lips partially obscured by microphones. These common occurrences significantly reduce the visibility of facial expressions and mouth movements, which are critical for training a lip reading model. Such examples can be seen in Figure 4.3.

For these reasons, I decided to use a different data source for my main dataset. I selected data from TV shows broadcasted by Czech Television. This program meets the conditions of the original English LRS2 dataset. There are longer sections where one person is talking to the

Figure 4.3 Sample from the Czech Parliament. Speakers look down or have partially hidden lips.

camera, filmed from the front view, so the lip region is clearly visible. I communicated with Czech Television about the conditions of their use. The data from their shows are available for academic use; however, they cannot be shared without permission.

I started with the TV show called Reportéři ČT. It is an investigative program that brings political, economic, and social cases from home and abroad. Each episode lasts about 35-45 minutes. There are about 40 episodes per year, about 500 episodes of broadcast between 2009 and 2024 were used as input data.

I could not continue with the same data source since the quality and the way of recording changed substantially in 2009. So I chose a different TV show called Černé ovce. It is an investigative journalism show dedicated to defending consumer rights. Reporters tackle pressing issues, conduct tests, and offer advice. Each episode lasts about 13-15 minutes. There are four episodes per week, and I used about 2500 episodes for my dataset.

The same problem arose, and I had to choose another program. The last TV show I selected is Zprávy ve 12. It is a news program that includes sports news and weather reports every weekday at 12.00. Each episode lasts about 20-30 minutes. I used about 3500 episodes.

Negotiating legal matters and permissions with Czech Television resulted in an agreement allowing me to share a portion of my dataset gathered from the TV show Zprávy ve 12 for academic, non-commercial purposes. Access to the remainder of the test dataset and the training dataset is available upon receiving individual approval from Czech TV.

4.1.2 Pipeline

The processing pipeline includes four steps:

- Downloading the data
- Splitting the video into sequences
- Identifying sequences in which lip movement is present
- Transcribing

The final output is sequences of video paired with a text file containing transcription. The diagram of this pipeline can be seen in Figure 4.4.

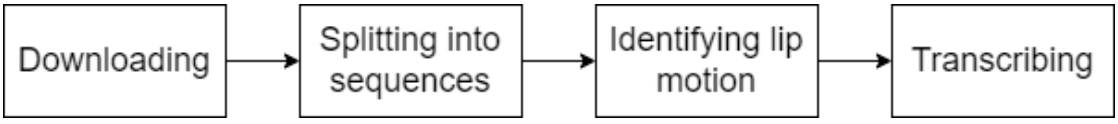

Figure 4.4 Diagram of dataset processing pipeline.

4.1.2.1 Downloading

For the downloading process, I used a command-line program `yt-dlp` [1]. It enables the user to download videos from Youtube and other video platforms. There was an issue when downloading videos from ČT using the latest stable version, so I had to use a different version instead.

4.1.2.2 Splitting into sequences

This step aims to split the videos into shorter sequences following the pattern of the LRS2 dataset. To achieve this, I extracted the audio from the input recording and detected occurrences of silence in it using a Python library called `pydub` [2]. When silence is longer than a given threshold, it indicates a new sentence or phrase, and the video is cut at this point. The mean duration of pauses between sentences in speech can vary significantly depending on language, speaking rate, speaker characteristics, and context. Research suggests that pauses between sentences typically range from about 0.2 to 0.6 seconds. In American English conversational speech, the pause is approximately 327 milliseconds [53]. News anchors and presenters typically speak at a slower pace than in an everyday conversation, ensuring that viewers can easily comprehend the information; therefore, after a few experiments, I set the pause threshold to 500 milliseconds.

Sequences shorter than 1 second and longer than 15 seconds are excluded from the dataset due to their unsatisfactory duration. When a sequence is too short, it does not contain enough information and context and can easily be misinterpreted. Too long sentences are harder to interpret.

4.1.2.3 Identifying lip movements

Data collected from the previous step still contain many recordings in which a person does not speak because the reporter speaks in the background to describe the event while an illustrative video is shown. These sequences where faces are absent or not clearly visible are inappropriate for lip reading. Therefore, only those recordings are selected where the mouth motion is identified.

This step is realized using face landmark detection based on iBUG [3] facial landmark standard. It identifies 68 key points that define the contours of the face. These points can be seen in Figure 4.5.

[1] `https://github.com/yt-dlp/yt-dlp`
[2] `https://github.com/jiaaro/pydub`
[3] Intelligent Behaviour Undestanding Group by Imperial College London. "The core expertise of the iBUG group is the machine analysis of human behaviour in space and time, including face analysis, body gesture analysis, visual, audio, and multimodal analysis of human behaviour, and biometrics analysis. Application areas in which the group is working are face analysis, body gesture analysis, audio and visual human behaviour analysis, biometrics and behaviometrics, and HCI." `https://ibug.doc.ic.ac.uk/`

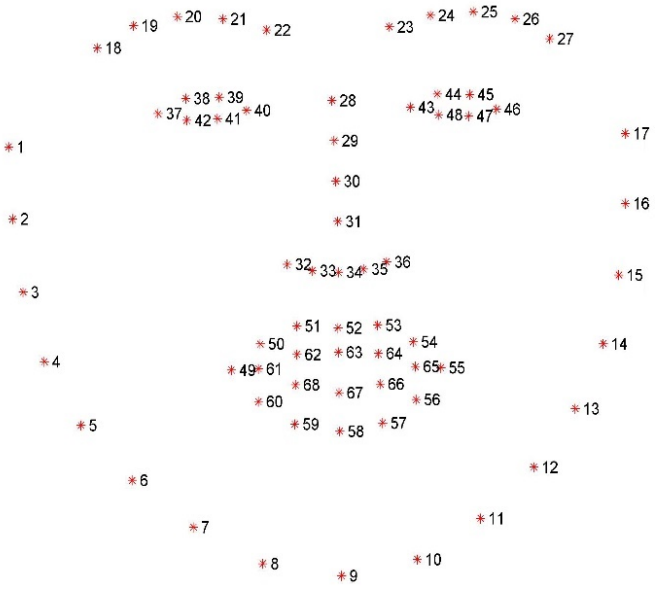

Figure 4.5 The locations of the 68 facial landmarks developed by iBUG.

People exhibit various patterns of mouth movement while speaking, reflecting individual articulation habits, linguistic differences, and emotional expressions. One common movement is the opening and closing of the mouth to articulate vowel and consonant sounds, with different degrees of mouth opening corresponding to vowel qualities. So, one way to detect whether a person is speaking is to observe if the mouth is opened. The iBUG landmarks $49 - 68$ outline the contours of the lip. These contours are shown in Figure 4.6.

In particular, the landmarks $61 - 68$ outline the inner lip. The distance d between y-coordinates of these landmarks, especially between 63 and 67 located in the center of the lips, can identify the speaking process. If the video clip contains frames where this distance is longer than a given threshold, it indicates speech. I experimentally set this threshold to $d = 3$. Examples of different distances between the inner lips can be seen in Figure 4.7.

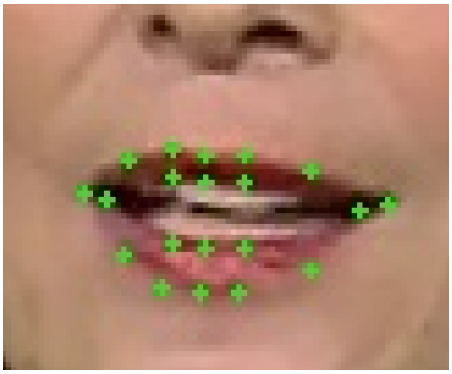

Figure 4.6 Landmarks $49 - 68$ outlining the contours of the lips.

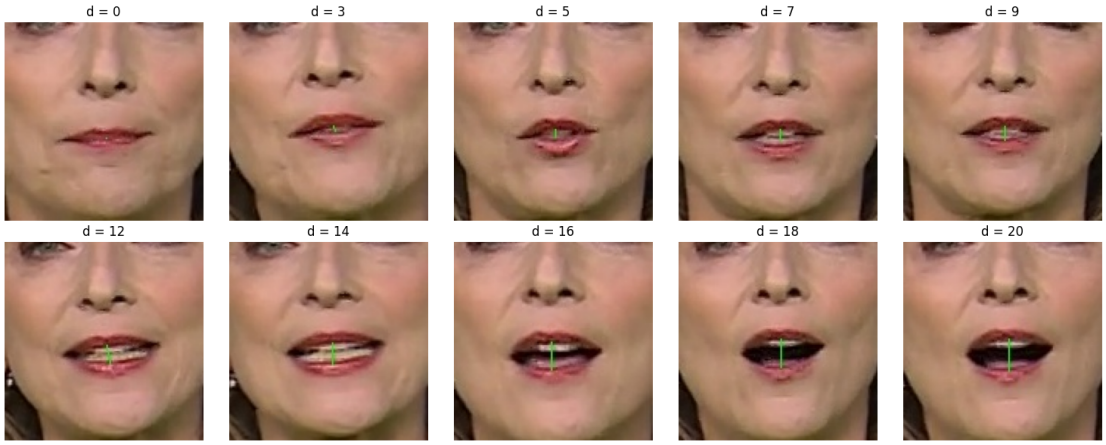

Figure 4.7 Different distance d between the lips during the speech.

When the camera is positioned further away from the recorded person, the apparent distance between the lips appears smaller because the mouth occupies less space within the frame. This opens up the debate about whether it would be more relevant to calculate some relative metric, such as the ratio of lip distance to total face area. However, clear articulation is essential to accurately distinguish the movements of the lips. Subtle lip movements filmed from a distance are difficult to recognize using the lip reading model. That is why I kept the distance as an absolute value, ensuring only videos with clearly visible lips are included.

The whole process of identifying the movements of the lips works as follows:

- An essential condition to check is if exactly one face is present. If this condition is not satisfied in any stage of the control, which means that more faces are identified, or no face is detected at all, the video is removed. I used `dlib` [4] library for face and landmark detection.
- At the beginning, the first five frames of each sequence are checked to see if the lip distance is longer than a threshold. Five frames are checked because when a person starts speaking, the lips may not be opened since the first frame. If this condition is not met, the video is removed.
- When the first criterion is satisfied, 5 % of the frames are analyzed in the same manner. The reviewed frames are evenly spread throughout the video; for example, if the sequence contains 100 frames, every 20th frame is checked. Controlling 5 % of the data is a reasonable compromise between speed and performance.
- If the lip distance is not long enough, the next three frames are checked. If any of these frames satisfies the distance, the control continues. This step treats the case when a person's mouth is closed as a pause between words.

After this process, only recordings with appropriate properties are left in the dataset. These conditions are strict, but they only include videos where the movement of lips is visible.

4.1.2.4 Transcribing

Transcription is the process of converting spoken words and dialogue from a video into written text. It can be achieved using automatic speech recognition (ASR) systems that analyze audio

[4] `https://github.com/davisking/dlib`

tracks and accurately transcribe spoken content. For this process, I used Google Cloud Speech-to-Text [5], a tool trained on millions of hours of audio data and billions of text sentences. It supports more than 100 languages, including the Czech language. Since this automatic transcription is not perfect and still has some error rate, it is convenient to manually check the obtained results and correct the mistakes.

4.1.3 Experiments

Naturally, transcripts obtained from automatic speech-to-text systems will not be as accurate as manually created transcripts. To see if this error is acceptable for lip-reading, I used test data from the LRS2 dataset that are annotated without mistakes and evaluated their performance on an English VSR model [21]. Then, I used the same videos, annotated them automatically and tested them on the same model.

The mean word error rate for original videos was $WER = 49.85\%$, and for automatically annotated $WER = 53.14\%$. The distribution of the error rate can be seen in Figure 4.8 and Figure 4.9. The word error rate of the automatically annotated data was slightly higher; however, it is still acceptable.

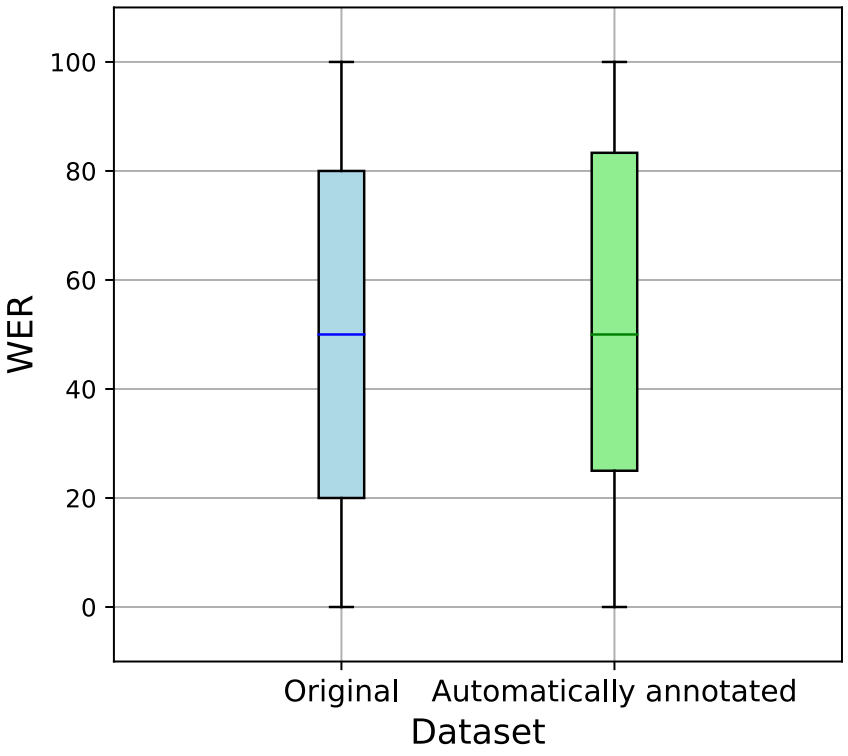

Figure 4.8 Boxplot of word error rate for original LRS2 test videos and automatically annotated videos.

[5] https://cloud.google.com/speech-to-text

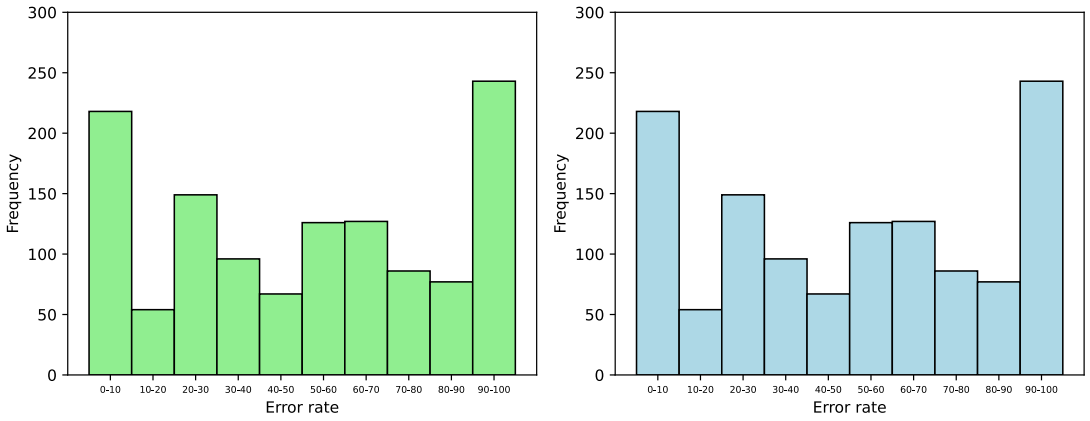

Figure 4.9 Histograms of word error rate for original LRS2 test videos and automatically annotated videos.

4.1.4 Final dataset

In this subsection, I provide some basic statistics about the created dataset and its parts. It can be seen that there are differences depending on the source. For example, data collected from Černé ovce is less diverse with a smaller vocabulary. Utterances from Zprávy ve 12 are significantly longer than from other datasets, having a richer vocabulary. The boxplot comparing the utterance length in words can be seen in Figure 4.10. Table 4.3 and Table 4.4 summarize all dataset statistics.

The final dataset I used is comprised of all of these three data sources. I will mention it in the text as `my_dataset`.

Dataset	# utterances	# total words	Average length	Vocabulary	Max length
Reportéři ČT	2,915	25,267	8.64	7,636	34
Černé ovce	2,350	22,456	9.56	6,607	32
Zprávy ve 12	2,561	33,706	13.16	10,992	37
My dataset	7,826	81,429	10.40	19,060	37
Parliament	2,469	21,591	8.74	5,987	38

Table 4.3 Summary of transcripts of the created dataset.

Dataset	Total duration in minutes	# frames	Average # frames	Average duration in seconds
Reportéři ČT	101.75	272,623	93.52	3.74
Černé ovce	149.58	224,363	95.47	3.82
Zprávy ve 12	225.60	338,413	132.14	5.29
My dataset	556.93	835,399	106.74	4.27
Parliament	151.50	227,250	92.08	3.68

Table 4.4 Summary of video data.

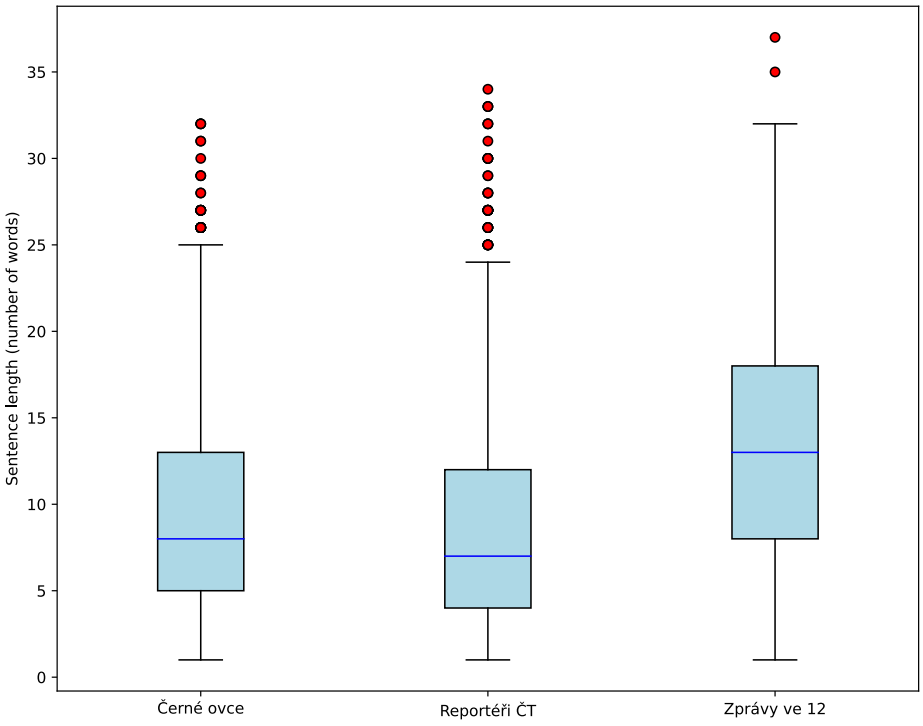

Figure 4.10 Boxplot of average sentence length of different data sources.

The train, validation and test split of `my dataset` is shown in Table 4.5.

Set	# files	Total duration in minutes
Train	7,042	500.90
Validation	392	28.22
Test	392	27.81

Table 4.5 Train, validation and test split of `my_dataset`.

4.1.4.1 Reportéři ČT

Figure 4.11 shows a sample taken from the dataset Reportéři. There are different speakers in different environments.

Figure 4.11 A sample from Reportéři ČT.

These are some illustrative utterances taken from this dataset:

```
Text: NEVINNÝ UMÍRÁ
Text: OBJASŇOVÁNÍ NAKOLIK
Text: TAKŽE JSEM MU VYSVĚTLOVAL O CO JDE ALE ON JE PŘESVĚDČENÝ ŽE
Text: TAKÉ NÁSLEDUJÍCÍ REPORTÁŽ JE ZE SEVERNÍCH ČECH A TÝKÁ SE PODEZŘENÍ Z KORUPCE
Text: ŽE TY ZEMI BUDOU S TAKY POSTIŽENÉ DŘÍVE NEBO POZDĚJI
Text: KDYŽ MĚ TŘEBA NA DRUHÝ DEN VIDĚL ŽE MÁM MONOKLY MODŘINY
Text: TAK BY MĚL PROSTĚ AKCEPTOVAT PRINCIP PARLAMENTNÍHO REŽIMU
```

4.1.4.2 Černé ovce

This dataset is also diverse in terms of visual settings. There are various speakers, but the anchor, Iveta Fialová, is often present. A sample is displayed in Figure 4.12.

Figure 4.12 A sample from Černé ovce.

These are some illustrative utterances taken from this dataset:

```
Text: TAKŽE NEVÍM JAK BY SE TO ŘEŠILO
Text: NEDOKÁZALI MI ŽE V DOBĚ KDY JSEM PARKOVAL
Text: PŘITOM NA TO MÁM DVA SVĚDKY
Text: A TAK MÍSTO TOHO ABY PRODEJ HRNCŮ VYHYNUL STÁVÁ SE Z NĚJ OBLUDNÝ PROBLÉM
Text: A ABYCH NEČINIL KROKY KTERÝCH BYCH POZDĚJI MOHL LITOVAT
Text: NAOPAK NEJDRAŽŠÍ BYLA MLETÁ KÁVA JIŽ PŘIPRAVENÁ DO KAPSY
Text: JSOU RODIČE KTEŘÍ POVAŽUJÍ VÝCHOVU SVÝCH DĚTÍ ZA PRVOŘADÝ ÚKOL BOHUŽEL
JSOU ALE I TAKOVÍ KTEŘÍ MAJÍ S RODIČOVSKÝMI POVINNOSTMI TROŠKU PROBLÉMY
```

4.1.4.3 Zprávy ve 12

These data are less variable in terms of the background – many videos are recorded in a TV studio and feature a TV news anchor. A sample is shown in Figure 4.13.

Figure 4.13 Example frames from videos from Zprávy ve 12.

These are some illustrative utterances taken from this dataset:

```
Text: A JE ZAJÍMAVÉ ŽE
Text: JEŠTĚ HODNĚ PRÁCE
Text: VŠUDE BYLA POPRVÉ A TOMU TAKÉ ODPOVÍDAL ZÁJEM OBYVATEL VE VĚTŠINĚ OBCÍ ŠLO
Text: K NĚJAKÉMU POKLESU DOJDE URČITĚ JEŠTĚ NA PODZIM
Text: PRAŽSKÉ BURZE SE I DNES DAŘÍ INDEXY AKTUÁLNĚ PŘIPISUJE PĚT DESETIN PROCENTA
Text: ZDRAŽILO HLAVNĚ BYDLENÍ NAOPAK ZLEVNIL ALKOHOL ÚDAJE ZVEŘEJNIL ČESKÝ
STATISTICKÝ ÚŘAD
Text: ÚSPĚŠNĚ ZAHÁJIL ČTVRTFINÁLOVOU SÉRII TAKÉ LITVÍNOV VERVA NA DOMÁCÍM LEDĚ
PŘEHRÁL A PŘEDEVŠÍM DÍKY DVĚMA PŘESILOVKOVÝ TREFÁM PARDUBICE 4 0
```

4.1.5 Invalid videos

There are some typical situations in which this processing pipeline had some issues. The first occurred mainly in data obtained from Reportéři. This show sometimes contains reports from abroad. These reports are either dubbed or contain subtitles. If the video was dubbed, the pipeline correctly created the sequences of the speaking person; however, it did not know that it is a different language and what is said does not match the movements of the lips. These videos are, therefore, unsuitable, and I had to remove them manually.

In the case of subtitles, an empty transcript is returned since transcribing is set to the Czech language. These videos are excluded automatically before the dataset is split into training, validation and test sets. The algorithm includes only videos with a transcript which is not empty.

Figure 4.14 Invalid videos in which the spoken language is not Czech. An empty transcript is returned, and the videos are excluded later.

The next issue that emerged was when two persons were having a dialogue, one from the front and the second from the side or behind. In this case, the face detector correctly detected only one face. But sometimes, the person in front was not speaking but still had facial expressions as a reaction to the other person speaking. In these situations, the facial reactions were evaluated incorrectly as speaking. Again, these videos are inappropriate, and I had to remove them manually.

Figure 4.15 Invalid videos in which facial expressions were classified as speech.

Chapter 5
Implemented models

After the research, I decided to adopt two architectures [27, 21]. Both models have a comprehensive architecture with 2D+3D Res-Net front-end and a back-end based on Conformer architecture [42]. Using the latest findings in the field of deep learning, the authors have gradually improved the architecture and training strategies. Their approach achieves one of the best results of all the models used in this domain [21].

The first one [27], which I call the simple model in this work, experiments with other languages such as French, Spanish, or Portuguese, which is rather exceptional since most of the research is focused on English, alternatively the Chinese language. However, the size of their dataset is substantially larger than the dataset I was able to collect. For example, for French they used 58 809 videos (84.9 hours) for training, 333 videos (0.4 hours) for validation and 235 videos (0.3 hours) achieving $WER = 67\%$ on the Multilingual TEDx-French test dataset and $WER = 59\%$ on the CMU-MOSEAS-French dataset. Collecting such a large dataset would be out of the scope of this work due to significant time constraints.

The second adopted model [21] experiments with automatically generated labels and has proven that enlarging an English dataset with automatically generated transcripts can increase the performance of the model. This inspired me to create my dataset using the automatic labels as well. In addition to their best model, which is trained on a huge dataset consisting of 3448 hours of data and achieving $WER = 20.3\%$, they have also trained models on smaller subsets. The one trained on the 23-hour LRS3 subset reaches $WER = 72.5\%$. This is more than twice the amount of my dataset but the difference between the error rate is not that significant.

5.1 Overall architecture

Both models are end-to-end trainable, which means that feature extraction and recognition are trained together. On a high level, they follow standard lip-reading architectures: an input, a video sequence (or batch of video sequences), is given to the feature extractor, which extracts the ROI area around the lips. This pre-processed video continues to the visual front-end based on 3D CNN and ResNet-18 as a sequence of images. This module maps the images to the visual speech representations that are fed to the back-end encoder that has a conformer architecture. The last module is a transformer-based decoder that decodes lip movements as ASCII characters. There is a linear layer on the top to progressively output sequence of these characters – a sentence, which is marked as finished when a special character $< eos >$[1] is appended. The high-level architecture of the implemented model can be seen in Figure 5.1.

[1] end of sequence

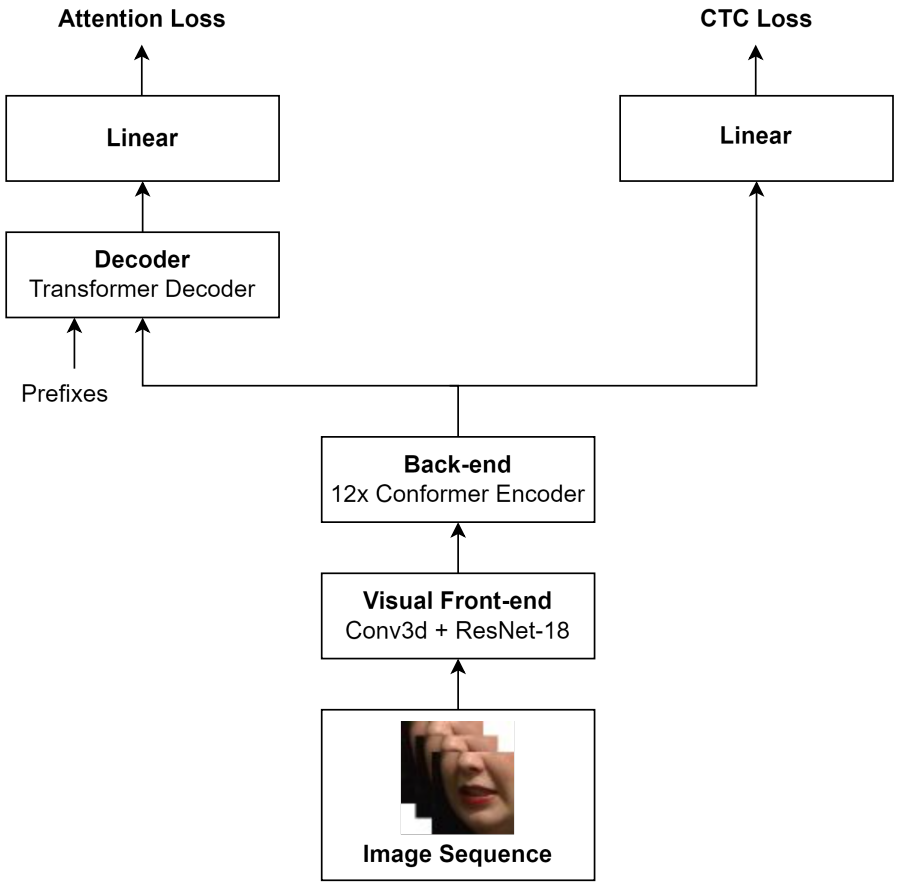

Figure 5.1 General architecture of implemented models.

5.1.1 Pre-processing

Utilizing the RetinaFace [54] face detector and the Face Alignment Network [55], 68 facial landmarks are identified. Faces are aligned to a neural reference frame to mitigate rotational and scaling differences through a similarity transformation. The ROI is extracted employing a bounding box of 96×96, positioned at the center of the mouth. Each frame is normalized by subtracting the mean and dividing by the standard deviation of the training set. The result of this process can be seen in Figure 5.2.

5.1.2 Front-end

The front-end [42] is the same for both models. It is a modified version of ResNet18 where the initial convolutional layer has been substituted with a 3D convolutional layer that has a kernel size of $5 \times 7 \times 7$. At the end of the residual block, the visual features are condensed across the spatial dimension using a global average pooling layer. The full architecture can be found in Appendix Section A.2.

Figure 5.2 A sample from pre-processed video.

5.1.3 Back-end

The back-end encoder utilizes Conformer encoder architecture consisting of 12 conformer blocks. The positional embedding component uses a linear layer to map the output from ResNet-18 into an *adim*-dimensional space. These transformed features are encoded with relative position information. The architecture includes a feed-forward module with a linear layer that elevates the features into a higher *edim*-dimensional space. This is followed by a Rectified Linear Unit (ReLU) activation, a dropout layer, and another linear layer that returns the output to an *adim* dimension. A multi-head mechanism with h heads uses various linear projections to scale down to an $edim/h$ dimension. The attention process is executed concurrently across each head, with the results merged back into an *adim*-dimensional space and projected to final values of size *odim*.

The convolutional module is composed of a 1D point-wise convolutional layer, Gated Linear Units [56], a 1D depthwise convolutional layer, a batch normalization layer, a swish activation layer, another 1D point-wise convolutional layer, and a layer normalization. The difference between the architectures is in parameters $adim, edim, h$ and $odim$.

5.1.4 Simple model

The parameter settings for a model with the simpler architecture are the following:

- $adim = 256$
- $edim = 2048$
- $h = 4$
- $odim = 41$, since the model is trained to predict characters from the English alphabet.

5.1.5 Complex model

The parameter settings for more complex architecture are the following:

- $adim = 768$
- $edim = 3072$
- $h = 12$
- $odim = 5048$, since the model is trained to predict unigram-based units from a vocabulary of size 5000.

5.1.6 Decoder

The decoder follows the standard decoder architecture described in 2.2.2. The model uses a beam search that tries to find a more optimal sequence by keeping multiple hypotheses at each step and extending them in parallel. At each step, it extends each prefix in the current set of best candidates by considering all possible next tokens (according to the model's prediction). hus, during the training procedure, besides the encoded sequence, prefixes of the target sequence are taken as input. These prefixes, ranging from index 1 to $l - 1$ where l is the target length index, are projected to embedding vectors.

5.1.7 Loss function

5.1.7.1 CTC

A Connectionist Temporal Classification loss [57] is useful in situations where sequence alignment is necessary but difficult to achieve. Such an example is speech; people have different speech rates, which causes input and output sequences to differ in length. CTC computes a loss between a continuous time series and a target sequence by adding all potential alignments between the input and target, yielding a loss value that is differentiable with respect to each input node. It was first introduced in lip reading when ASCII characters were used as units of classification.

For an input sequence $X = [x_1, x_2, \dots x_M]$ of size M an output sequence $Y = [y_1, y_2, \dots y_N]$ of size N is predicted. The goal is to find the most probable sequence $\hat{Y}$. CTC loss assumes conditional independence between each output prediction.

$$\begin{aligned} P_{CTC}(\boldsymbol{y} \mid \boldsymbol{x}) &\approx \prod_{m=1}^{M} p(y_m \mid \boldsymbol{x}) \\ \mathcal{L}_{CTC} &= -\log P_{CTC}(\boldsymbol{y} \mid \boldsymbol{x}) \end{aligned} \tag{5.1}$$

5.1.7.2 Attention loss

An attention-based model directly estimates the posterior on the basis of the chain rule.

$$\begin{aligned} P_{att}(\boldsymbol{y} \mid \boldsymbol{x}) &\approx \prod_{n=1}^{N} p(y_n \mid y_{<n}, \boldsymbol{x}) \\ \mathcal{L}_{att} &= -\log P_{att}(\boldsymbol{y} \mid \boldsymbol{x}) \end{aligned} \tag{5.2}$$

5.1.7.3 Hybrid Attention/CTC

Attention-based methods perform alignment between frames and recognized symbols, while CTC uses Markov assumptions to solve sequential problems. A hybrid CTC/attention [58] utilizes both architectures to improve robustness and achieve faster convergence. The final objective is:

$$\mathcal{L} = \alpha\mathcal{L}_{CTC} + (1 - \alpha)\mathcal{L}_{att} \tag{5.3}$$

where α is a tunable parameter that controls the relative weight of CTC and attention mechanisms.

The attention objective is approximated letter-wise, while the CTC is a sequence-level objective.

5.1.8 Performance metrics

Common performance metric in speech recognition models is the character error rate (CER). It measures the "distance" between ground-truth transcription and predicted text. It is defined as

$$\mathrm{CER} = \frac{\mathrm{S} + \mathrm{I} + \mathrm{D}}{\mathrm{N}} \tag{5.4}$$

where S is the number of character substitutions, I is the number of insertions, D is the number of deletions needed to get from hypothesis to the reference sequence and N is the total number of characters in the target sequence.
Similarly, a word error rate measures how close the predicted word sequence is to the target word sequence:

$$\mathrm{WER} = \frac{\mathrm{S} + \mathrm{I} + \mathrm{D}}{\mathrm{N}} \tag{5.5}$$

S, D and I are computed at the word level and N is the total number of words in the target sequence.

Chapter 6

Model improvements

The models were implemented using PyTorch Lightning [1]. It is an open-source Python library providing a high-level interface for PyTorch. It organizes PyTorch code, making deep learning experiments easier to read and reproduce. It is suitable for large and complex models. Configuration *yaml* files are used to simplify the structure of different models. These files contain information about the architecture, training strategy, hyperparameters, paths to the dataset and others. A detailed explanation of these files is included in the source code. All models were trained on the remote server with two NVIDIA GeForce RTX 2080 Ti GPUs. These resources were provided by Profinit.

The models were trained in 30 epochs. This number was determined from the train and validation curves shown in Figure 6.2. After 30 epochs, the model reached a point where the validation loss converged and ceased to show improvement, instead beginning to increase slightly. The top two checkpoints based on the WER on the validation set after each epoch are systematically saved during the training process. This approach ensures that the model versions with the lowest WER are preserved for further evaluation and testing. The evaluation is then performed on the average of these two checkpoints combined with the last checkpoint.

The model is trained using AdamW [59] optimizer with a cosine learning rate scheduler with five warm-up epochs. The peak learning rate is set to 0.001, as suggested by the authors of the models. Figure 6.1 shows the learning rate curve during the training process. Experimenting with the hyperparameters is provided in Chapter 7 Section 7.2. The original implementation uses a batch size of 1800 frames. It is suggested to use the maximum number of frames that can fit into the memory, which was 1000 in my case.

[1] PyTorch Lightning

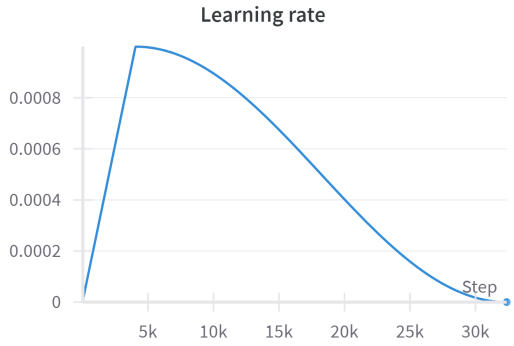

Figure 6.1 Scheduling of the learning rate during the training process.

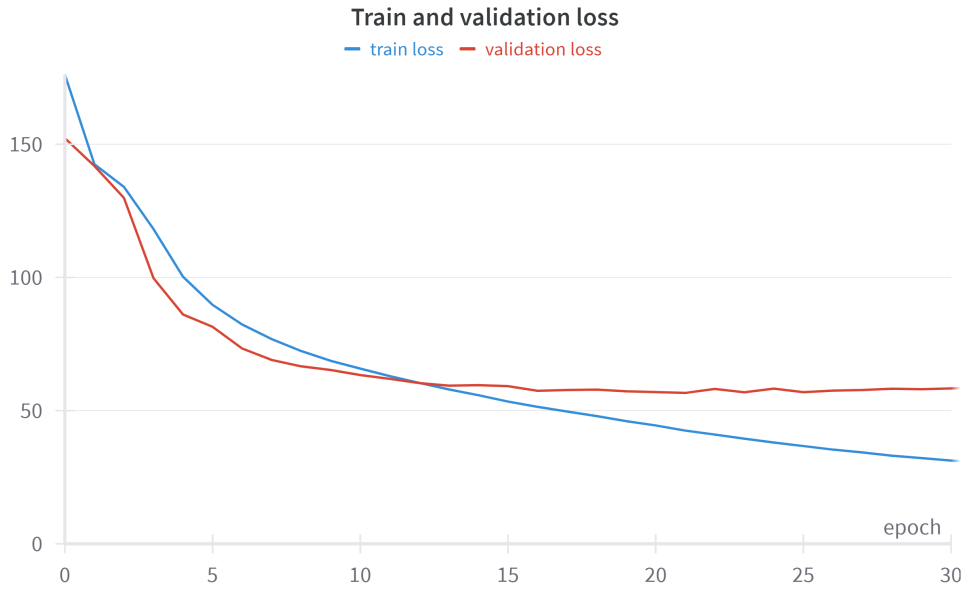

Figure 6.2 Train and validation loss during the training of simple model with diacritics on 30 epochs.

I use two English pre-trained models:

1. Simple model trained on the LRS2 dataset, achieving $WER = 26.1\%$.
2. Complex model trained on 3448 hours of data, achieving $WER = 20.3\%$.

6.1 Character-based model without diacritics

There is no similar lip-reading model for the Czech language; the naive approach was to use an English pre-trained model for my data. This model is trained to output 41 classes – English letters, numbers, white space and a symbol for an unknown character. Without any changes this model can predict only without diacritics. The output classes are the following list of ASCII characters:

```
"<blank>", "<unk>", "'", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
"<space>", "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M",
"N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z", "<eos>"
```

The pre-trained model was trained in the English language, so it makes predictions in the English language as well. However, it is not very reasonable to measure any metrics. The WER and CER can be higher than 100% if the hypothesis is longer than a transcript. This method reached the mean $WER = 133\%$ and $CER = 97\%$.

What is more meaningful is to have a look at the predictions in terms of subjective evaluation, which mathematical metrics cannot capture. It can be seen that some words in Czech are mapped to English words that sound and look similar. These words share the same visemes, and the model that was taught in English outputs an English equivalent that is phonetically close to the Czech word.

```
hypothesis:
I LIKE TO MUSIC MORNING AND PUT THE
reference:
A JA TO MUSIM UDELAT PROTI
```

In this case the pronunciation of word `I - /ai/` is close to Czech `A JA`. The word `MUSIC` was predicted instead of `MUSIM`.

```
hypothesis:
IF WE HAVE TYPE OF PASTA I'D LIKE TO TURN OUT A HALF MOST OF IT
reference:
KVULI HORKU VCERA POPRASKALA DRAHA NA LETISTI LEOSE JANACKA V MOSNOVE
```

In this utterence the end of word `TYPE` together with `OF PASTA I` were trying to match word `POPRASKALA`. The word `MOSNOVE` was predicted as `MOST OF IT`.

In longer sequences, the equivalent mapping is not that obvious, for example:

```
hypothesis:
AND WITH THE MACHINE A COUPLE OF DAYS NOT ONLY SEEM
REALLY BASICALLY COMING FOR TOP OF THE DAY
refrence:
SAMOZREJME ZE TAKOVY TEN KDO TO KDYSI UDELAL A OBEC
JAKO BY O TOM NEVEDELA
```

6.1.1 Training on Czech subset

The next step was to keep the pre-trained model and train it on Czech data to see if it could learn to predict the Czech language. The first experiments were carried out on a smaller subset of dataset `Reporteri`. The model had $WER = 103.14\%$ and $CER = 72.25\%$.

```
hypothesis:
VELKEM SE SPOSTRATOVALI SE MAFIE
reference:
VIETNAMSKE ZPROSTREDKOVATELSKE MAFIE

hypothesis:
NETOPROVOLNE ZADOVAT NEKOMUNA SE TO VEC KONCEL
reference:
NEDOBROVOLNE ZAROVEN TEDY TA KOMUNIKACE V TU CHVILI SKONCILA

hypothesis:
KDYZ JSEM NAM TO ZILA VOTO PALA
reference:
KDYZ JSEM TAM DORUCILA FOTOAPARAT

hypothesis:
TOVOLENCE OBJEDNES
reference:
DOVOLAT SE OBJEDNAT

hypothesis:
TO ZMINOU LIKU A SE DALI PO ZE 20000
reference:
UZ JSME TUHLE TU AKCI DELALI V ROCE 2003
```

Now, the output was not in English but in some strange made-up language with a few English words. The model followed the pattern as before, trying to come up with a prediction which sounds or at least looks similar. In all the examples mentioned, the auditory resemblance is indisputable. This measurable progress led me to decide that this method could get better when having more data, and I started to collect more data.

When I plotted the error rate histogram, there was an outlier with $WER = CER = 0\%$. When I looked up that sentence, it was this:

```
hypothesis:
ZE DOSLO K VYBUCHU V DUSLEDKU NEODBORNE MANIPULACE
reference:
ZE DOSLO K VYBUCHU V DUSLEDKU NEODBORNE MANIPULACE
```

I was very surprised; I expected an easy sentence with one word that could possibly be predicted with 0 error rate. The reason behind this was that this was a statement of former president Miloš Zeman, and it was broadcast in more episodes of the show over the years when necessary for the report's context. This video was presented three times in the training set and one time in the test set, thus predicted without any mistakes.

6.1.2 Final models

The final simple model without diacritics reaches $CER = 48.4\%$ and $WER = 80.8\%$. The performance of the complex model is $CER = 47.5\%$ and $WER = 84.8\%$.

6.2 Character-based model with diacritics

English belongs to the Germanic language group within the Indo-European family. Czech is a West Slavic language. These two languages do not share many linguistic elements.

The Czech alphabet introduces several unique characters that differentiate it from the English alphabet, adding layers of complexity and richness to the language. In addition to the standard 26 letters found in English, Czech includes accented vowels and other special characters, making a total of 42 letters. Characters such as “č,” “ř,” “š,” “ž,” which are not found in English, incorporate a diacritical mark called caron (háček). This mark modifies the pronunciation, adding sounds that are specific to Czech and other Slavic languages. Vowels can bear acute accents, such as “á,” “é,”, “í,” “ó,” “ú,” and “ý,” which indicate a longer pronunciation duration. Additionally, the ring diacritic appears exclusively on ”ů,” further distinguishing Czech orthography. These special characters convey accurate pronunciation and meaning, reflecting the phonetic nuances intrinsic to the Czech language.
The character-based model with diacritics, therefore, predicts the following 57 classes:

```
"<blank>", "<unk>", "'", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
"<space>", "A", "Á", "B", "C", "Č", "D", "Ď", "E", "É", "Ě", "F", "G", "H",
"CH", "I", "Í", "J", "K", "L", "M", "N", "Ň", "O", "Ó", "P", "Q", "R", "Ř",
"S", "Š", "T", "Ť", "U", "Ú", "Ů", "V", "W", "X", "Y", "Ý", "Z", "Ž", "<eos>"
```

The number of output classes is different from the pre-trained models, so I had to change the dimension of the architecture. I adopted two approaches:

1. Changing the last linear layer and corresponding layers in the decoder and CTC to the desired number of classes (57).
2. Adding a linear layer with the new dimension (57) on top of the original linear layer, which has 41 classes. The idea behind this is that model can benefit from already learned representations of English letters. The additional layer then extends the number of output classes by the special characters present in the Czech alphabet. This was only applied to the simple model since the original architecture of the complex model uses unigram tokenization with more than 5000 classes.

Changing of the output dimension required changing the following layers:

- Input dimension of embedding to 57. A new linear layer was initialized, so the model had to learn new embedding representations.
- Output dimension of linear CTC layer to 57.
- Output dimension of linear layer on the top of the decoder to 57. The old linear layer was either replaced or one more layer was added on the top.

6.2.1 Simple model

The simple model with the replaced linear layer achieved $CER = 47.3\%$ and $WER = 84.8\%$. As mentioned before, the error rate does not capture all the information about the predictions. For example, this sentence:

```
hypothesis: MIMO O TEM ZA TÝDNE
reference: MIMOCHODEM TATÍNEK
```

has high $WER = 250\%$ due to a longer predicted sequence than the target sequence. However, when we look at the prediction, words `MIMO O TEM` are almost homophones with the word `MIMOCHODEM`. `ZA TÝDNE` is close to `TATÍNEK`.

The next example shows how the model adapted to the Czech alphabet. An English name JEREMYHO is predicted as ČEREMI DO.

```
hypothesis: PO POLISMU SE PROTI EVROPSKÉ NÁHLADY JSOU PORÁŽEK V ABRICKÝCH
KRAMERCŮ ČEREMI DOHROMINA VELKÝM NEBEZPEČÍM PRO EVROPU

reference: POPULISMUS A PROTIEVROPSKÉ NÁLADY JSOU PODLE ŠÉFA BRITSKÝCH
LABOURISTŮ JEREMYHO CORBYNA VELKÝM NEBEZPEČÍM PRO EVROPU
```

This sentence

```
hypothesis: KOLIČNĚ ŽE SKUTEČNĚ V JAKÉKOLO JEŠTĚ ČEKAL TERNATIVÁN
reference: EKOLOGIČTĚJŠÍ A SKUTEČNĚ V TĚCH EKOLOGIČTĚJŠÍCH ALTERNATIVÁCH
```

demonstrates the problem of correctly determining the spaces between words. JAKÉKOLO JEŠTĚ ČEKAL TERNATIVÁN contains sequence ÉKOLOJEŠTEČEK that represents word EKOLOGIČTĚJŠÍCH and ALTERNATIVÁN word ALTERNATIVÁCH.

The second model with the added linear layer achieved $CER = 48.5\%$ and $WER = 87.3\%$.

6.2.2 Complex model

The complex model reached $CER = 46.8\%$ and $WER = 84.1\%$. For visual comparison, I show the same sentences as before:

```
hypothesis: MY MOHLO TAM ZADÍNE
reference: MIMOCHODEM TATÍNEK

hypothesis: PO POLICIMUS A PROTI EVROPSKÉ NÁKLADY JSOU PODAŘILY V ABRICKÝCH
LÍBILISTŮ ČEREMI DOCHORMÍ A VELKÝM NEBEZPEČÍM POHEVROPU

reference: POPULISMUS A PROTIEVROPSKÉ NÁLADY JSOU PODLE ŠÉFA BRITSKÝCH
LABOURISTŮ JEREMYHO CORBYNA VELKÝM NEBEZPEČÍM PRO EVROPU

hypothesis: KOLOGYČNĚ ŽE A SKUTEČNĚ V JAKÉ TOHO JEŠTĚ ČEKAL TERDA TIVÁ
reference: EKOLOGIČTĚJŠÍ A SKUTEČNĚ V TĚCH EKOLOGIČTĚJŠÍCH ALTERNATIVÁCH
```

The results show, that complex model reaches similar error rate. However, when we look at the sentences, the hypotheses resemble the reference less than in case of simple model and it more difficult to capture the meaning of the utterence.

6.3 Unigram-based model

Unlike the previous approach, where every character of the transcript was encoded separately, unigram language modelling [60] creates a vocabulary of subwords. I used SentencePiece model, an unsupervised text tokenizer and detokenizer. It handles the mapping of vocabulary to IDs and can directly produce sequences of vocabulary IDs from unprocessed sentences. This vocabulary list has to be diverse enough to sufficiently capture varying subwords while avoiding redundant information. It is language-independent – sentences are treated as sequences of Unicode characters without incorporating any language-specific rules. SentencePiece model successfully addressed the problem of unambiguity in tokenization when there are multiple ways to split up the word based on the vocabulary list using the subword regularization [60].
Example of tokenization:

```
TEN JAK VÍME PAK DOPORUČIL ZBAVIT IMUNITY JAK ANDREJE BABIŠE

3968  1227  4246  1158  2615     737    1152   4811   3902  1130  1889  2253
_TEN  _JAK   _V    ÍME  _PAK  _DOPORUČ   IL   _ZBAVI   T    _I     MU    NI

4091   1227     163    840    232    840
 TY    _JAK   _ANDREJ   E   _BABIŠ    E
```

_ symbols represent whitespace symbols that are replaced during tokenization.
The SentencePiece model was trained on the text input which consists of all transcripts from the train set with the vocabulary set to 5000.

The training-validation loss, seen in Figure 6.3 indicates the following:

- Train loss consistently decreases as the number of epochs increases, indicating that the model is learning and improving its performance on the training dataset.
- However, the validation loss decreases until epoch 7, then it starts to plateau and slightly increases, which suggests the beginning of overfitting, where the model learns patterns specific to the training data, reducing its generalizability.

Results of this model trained on 8 epochs as suggested by the plots of the loss function are $CER = 85.1\%$ and $WER = 100.0\%$. Although the word error rate of the model might not appear significantly worse at first glance, a deeper investigation into the character error rate reveals a high discrepancy. This model does not make any meaningful predictions. The sentences include repetetive words and sequences of characters. The primary cause of this training inadequacy is the insufficient volume of training data provided to the model. With an increasing number of output classes, even more training data are required. These are some examples predicted by this model:

```
hypothesis: A MY JSME SE PTALI ŽE JE TO
reference: JE ROZESTAVĚNÝCH FOTOVOLTAICKÝCH ELEKTRÁREN POPŘÍPADĚ TEPELNÝCH
ČERPADEL

hypothesis: JÁ SI MYSLÍM ŽE
reference: ŽE VLASTNĚ BUDOUCNOST TOHO PLÁNU

hypothesis: JÁ SI MYSLÍM ŽE
reference: NESTAČÍ ŽE JÁ VÁM BUDU RADIT JAK TO MÁTE STŘÍDAT
```

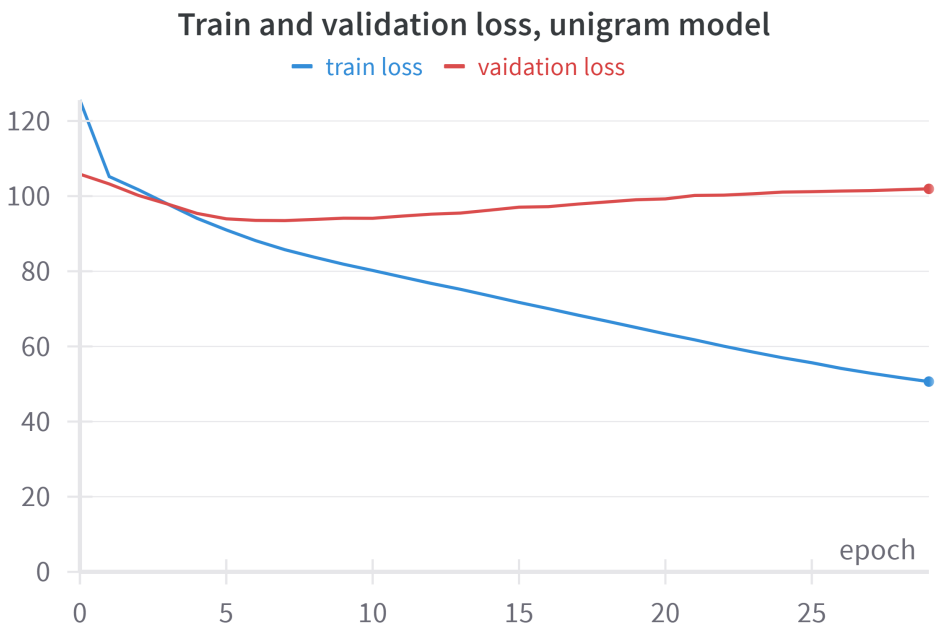

Figure 6.3 Training and validation loss for simple unigram model.

Chapter 7

Experiments and evaluation

7.1 Influence of the training set size

These experiments assess the influence of the training set size on the model's performance. It gradually changes the size of the train set while keeping other variables constant. The performance of each model is evaluated on the same validation set. Figure 7.2 displays the behaviour of validation loss for different training set sizes. Table 7.1 summarizes the performance metrics on each subset.

Figure 7.1 Validation loss for different training set sizes.

Train size	CER [%]	WER [%]
25%	57.8	98.0
50%	52.5	92.7
75%	49.9	89.5
100%	47.3	84.8

Table 7.1 Influence of the train set size on the performance.

These results demonstrate that with the growing number of training data, the performance of the model increases. The results suggest that the model's effectiveness is limited by the data available for its training.

Since the same validation set was consistently employed throughout the experiments, it is possible to analyze the progressive improvements in the predictions made. The following examples show how the sentence was refined and evolved when processed through various models.

```
reference: JSEM SI NACPAL VŠAK VÍM
0.25: PODLE TAK JSEM SI NA SMĚL VŠANÝM
0.5: A TO BYLO MOŽNOU NENÍ JSEM SI NA ZMEN VČERNÝ
0.75: JAK JSEM SI NA SPEK VŠECHNÝM
100: TO JSEM SI NASPĚL VŠECHNY


reference: NÁSLEDUJÍCÍCH DVOU MĚSÍCÍCH ZAMÍŘÍ DO VŠECH KRAJŮ A ZAČNE UŽ
PŘÍŠTÍ ÚTERÝ V OSTRAVĚ

0.25: NA VLÁSTARUJSE I DVOU JESTNĚ SE MÍŘÍTO VŠAK HRANU A ZAČAL UŽ
PŘÍŠTÍ UTRAJÍ V ORSTRAVĚ

0.5: NA VLÁSTITU SE DVOU MĚSÍCÍCH ZEMÍŘÍ TO VŠECH RAKU A ZA TO AUŽ
PŘÍŠTÍ ÚTRÉ V ODSTRAVIA

0.75: DRAVLÁSTIUJÍ SE DVOU MĚSÍCÍCH ZEMÍŘÍ TO VŠECH RADU HEZEČNĚ UŽ
PŘÍŠTÍ ÚTERÝ V OSTRAVIĚ

100: NA VLÁSLEDUJÍCÍCH DVOU MĚSÍCÍCH ZAMÍŘÍ TO VŠECH KRADU A ZAČNE UŽ
PŘÍŠTÍ ÚTERÉ V OSTRAVĚ


reference: ÚSTAVNÍ SOUD V BRNĚ BUDE ODPOLEDNE ROZHODOVAT O STÍŽNOSTI
ROMA JAROSLAVA SUCHÉHO KTERÝ PŘED NĚKOLIKA LETY ZAŽALOVAL MINISTERSTVO
ŠKOLSTVÍ KVŮLI DISKRIMINACI VLASTNÍ OSOBY

0.25: USLAVÍCOU V PRTÉ BUDE OD POLEDNEHO DOHODOVAT OSTÍČNOSTI NO
MARIONSKÉ VESTUJE HO KTERÝ PŘEDNÉHOLIKA LETECH SEČALOVÉ MINISTERSTVO
DOKOLO SVÝCH VŮLICH STÉMĚSTVACÍ HOSOBY

0.5: ÚSLAVNÍKŮ V PARTÉ BUDE ODPOLEDNE ROZHODOVAT OSTIŽNOSTI DO
MAJEROSLAVA SUCHÉHO KTERÝ PŘED EKOLIKA LETECH SYŠALOVÁ MINISTERSTVON
KOLS VNĚKVŮLI CEJÍM NA SI VLASTNÝCH OSROBY
```

```
0.75 : ÚSLAVISKOU UPERTÉ BUDE ODPOLEDNE ROZHODOVAT OSTŘEČNOSTI HO
PARAKOSLAVA SURNÉHO KTERÝ PŘED NÉHO LIDÉ LETY SEŽALOVAL
MINISTERSTVONDOLOSTVÍ KVŮLI TISKÝM NA SEVERENCÍ OSOBY

100: STAVNÍ SOUD V PRTĚ BUDE ODPOLEDNE ODCHODOVAT OSTŘEŠNOSTI HO
MAJELOSLAVA SUNÉHO KTERÝ PŘED NĚKOLIKA LETRISTIČALOVÉ MINISTERSTVO ŠKOLS
VY KVŮLI NICH MINA SIVNANTNÍ OSOBY
```

7.2 Hyperparameters

Adjusting hyperparameters is essential for enhancing the performance of machine learning models. The studies I used had thoroughly optimized these parameters, and I utilized their recommended settings. Nevertheless, I identified a few hyperparameters that could improve model performance.

7.2.1 Attention dropout

Since the models showed marks of overfitting, I experimented with attention dropout. Attention dropout is a regularization technique that is used during the training of transformer models. It is applied to the weights in the attention mechanism of transformers. By applying dropout to these attention weights, the model is encouraged to avoid relying too heavily on any particular data part, thus promoting more robust learning of dependencies across the input sequence.

The recommended setting [27] for the attention dropout rate is 0.1. I tried to increase the value to 0.3. The results evaluated on the validation set, displayed in Table 7.2 show that increasing this value did not enhance the model's generalization capabilities and did not perform better on unseen data.

Attention dropout	CER [%]	WER [%]
0.3	48.2	85.6
0.1	47.3	84.8

Table 7.2 Influence of the attention dropout rate.

7.2.2 Learning rate

The recommended setting for the learning rate is $lr = 0.001$. I experimented with different settings to see if they could positively affect the training procedure. The results are shown in Table 7.3. Figure 6.1 displays the difference of validation loss for different learning rates.

Learning rate	CER [%]	WER [%]
0.0003	50.0	90.0
0.001	47.3	84.8
0.01	82.7	98.9

Table 7.3 Influence of the learning rate.

Figure 7.2 Validation loss for different training set sizes.

The learning rate set to 0.0003 shows a steady and gradual decrease in loss over the 30 epochs, suggesting that the learning rate is helping the model improve smoothly without major fluctuations. When $lr = 0.01$ the plot indicates that the learning rate might be too high, causing the model to overshoot the optimal weights after the initial improvements. Setting the learning rate to $lr = 0.001$ also shows a consistent decrease in loss, similar to the $lr = 0.0003$ but at a slightly faster rate and achieving lower values. It suggests a balanced learning rate that allows the model to efficiently learn without the instability observed with a higher learning rate.

7.3 Excluding the numeric data

Numeric values can be a significant source of error due to their variable interpretations and contexts. For instance, numbers may represent dates, quantities, or identifiers and distinguishing between these uses can be challenging. There is an inconsistency in how numbers are represented – some are written as numerals, while others are spelled out in words. This variability can cause difficulties in accurately processing and understanding textual data. For example, the numeral "50" and the phrase "fifty" represent the same quantity but are expressed differently. This inconsistency also occurred in the dataset I collected:

```
TY SE MÁŠ TY JSI VYHRÁL TY JSI TEĎ V BALÍKU A JÁ MUSÍM SE VYSVĚTLOVAT
ŽE TO NENÍ PRAVDA ŽE MI VISÍ NA KRKU STOTISÍCOVÝ

Z 930 KUSŮ NEBO Z TISÍCE KUSŮ BUDOUCÍCH TRAMVAJÍ V PRAZE COŽ JE 25 PRO

PŘES 600 TISÍC ZAPLATIL PETR KREJČIŘÍK ZA PŘESTUP SVÉHO
ČTRNÁCTILETÉHO SYNA DO JINÉHO HOKEJOVÉHO KLUBU
```

These assumptions led me to experiment on whether it makes sense for future work to focus on replacing all numerals with their word equivalents. Instead of manually replacing all these numbers, I excluded all transcripts containing numerals. The size of the dataset in terms of utterances dropped by 12%.

The results, shown in Table 7.4 indicate that these numeric values are not a source of error in my case. The character error rate is slightly lower, but the word error rate is even higher than in a standard dataset. This is caused by the reduction of input data. The model benefits from every training example, even with inconsistent number representation. Thus, experimenting with the data format may be the case only when the model has reached the maximum accuracy, which cannot be further improved by adding more data.

Model	CER [%]	WER [%]
Simple model without numerals	46.9	85.8
Standard simple model	47.3	84.8

Table 7.4 Results on model with and without numerals.

7.4 Curriculum learning

In order to achieve better convergence in complex models, curriculum learning-based strategies can be applied. These techniques structure the learning procedure so that tasks or training samples become increasingly complicated over time. It draws inspiration from how people and animals learn, beginning with basic activities and working their way up to more difficult ones as their abilities advance. Curriculum learning can also be applied in lip reading by controlling the problem's difficulty. The model is trained on a subset of shorter utterances, gradually increasing the length of the sentences to the maximum.

Figure 7.3 shows the histogram of the length of videos counted in frames. Based on this distribution, I:

- Split the dataset into two parts: one containing videos whose length is less than 100
- Trained the simple model on short sequences for 15 epochs
- Trained this model on all sentences for 15 epochs
- Evaluated on standard test dataset used in this work

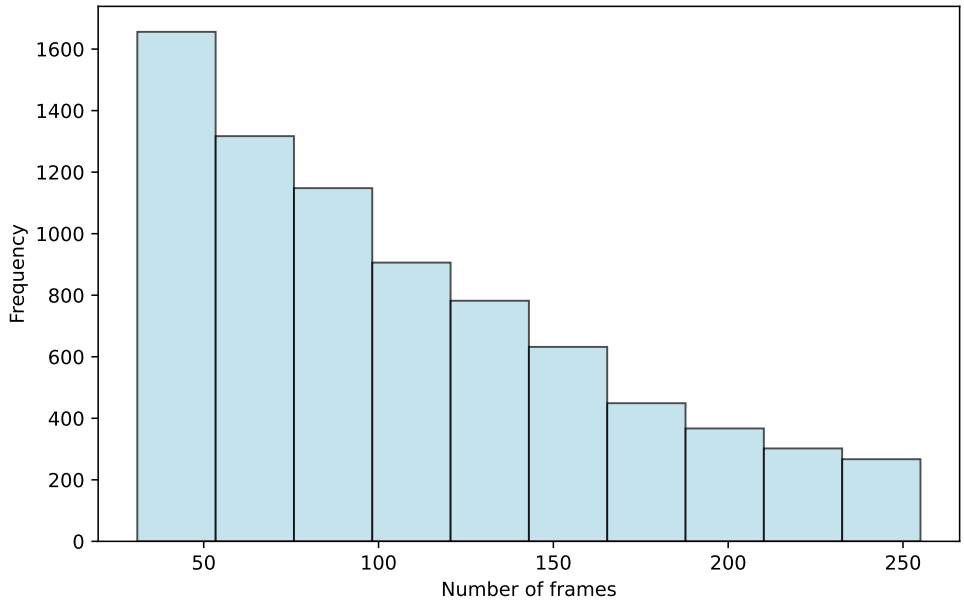

Figure 7.3 Histogram of the number of frames.

This method did not bring the desired improvement. Results can be found in Table 7.5.

Model	CER [%]	WER [%]
Simple model - curriculum learning	50.6	88.9
Standard simple model	47.3	84.8

Table 7.5 Results of the simple model using standard learning and curriculum learning.

7.5 Comparison of different data sources

In this experiment, I aimed to evaluate the performance of various models based on their training data sources and their behaviour when tested on specific subsets. Initially, I used the standard model trained on the training set. This model was then evaluated using a subset of the standard test set, where each subset comes from one data source. Subsequently, I trained separate models specifically on different data sources to investigate how the origin of training data influences model performance. Each of these models was then evaluated on the same test subsets to maintain consistency in comparison. The results of these evaluations are detailed in Table 7.6, which illustrates the performance metrics for each model across the different train and test subsets.

Train set	Test set	Simple model		Complex model	
		CER [%]	WER [%]	CER [%]	WER [%]
My dataset	Reportéři ČT	51.8	90.7	51.9	91.3
	Černé ovce	44.1	81.8	41.4	74.4
	Zprávy ve 12	45.0	81.6	49.0	86.4
Reportéři ČT	Reportéři ČT	59.7	100	57.8	98.9
Černé ovce	Černé ovce	50.1	92.4	49.3	91.7
Zprávy ve 12	Zprávy ve 12	46.4	88.4	44.6	84.7

Table 7.6 Performance comparison using different train and test subsets.

Generally, CER and WER are slightly better or comparable in the complex model across different combinations compared to the simple model. This suggests that added complexity in the model might help capture nuances in the data that the simple model misses. The differences in error rates, while measurable, do not translate into a substantial enhancement in capturing the real meaning of the conversations being analyzed. While the complex model may statistically outperform the simple one, the practical implications of this superiority in terms of understanding and processing human language are not that evident.

There are noticeable differences in performance depending on the training and test set combinations. There is a remarkable difference in performance when models are trained or tested with the `Reportéři ČT` dataset. Particularly, the simple model reaches a $WER = 100\%$ and complex model $WER = 98.9\%$ when both trained and tested on this dataset. It indicates that this dataset poses significant challenges that the models fails to handle.

7.6 Final evaluation

Final evaluation on the full test dataset. Results are shown in Table 7.7. Figures 7.4, 7.5, 7.6, 7.7 represent the boxplot comparison of WER or CER using different models.

Model	CER [%]	WER [%]
Simple model no diacritics	48.4	80.8
Complex model no diacritics	47.5	84.8
Simple model with diacritics	47.3	84.8
Complex model with diacritics	46.8	84.1
Simple model with added layer	48.5	87.3
Complex model unigram	85.1	100.0

Table 7.7 Final performance evaluation.

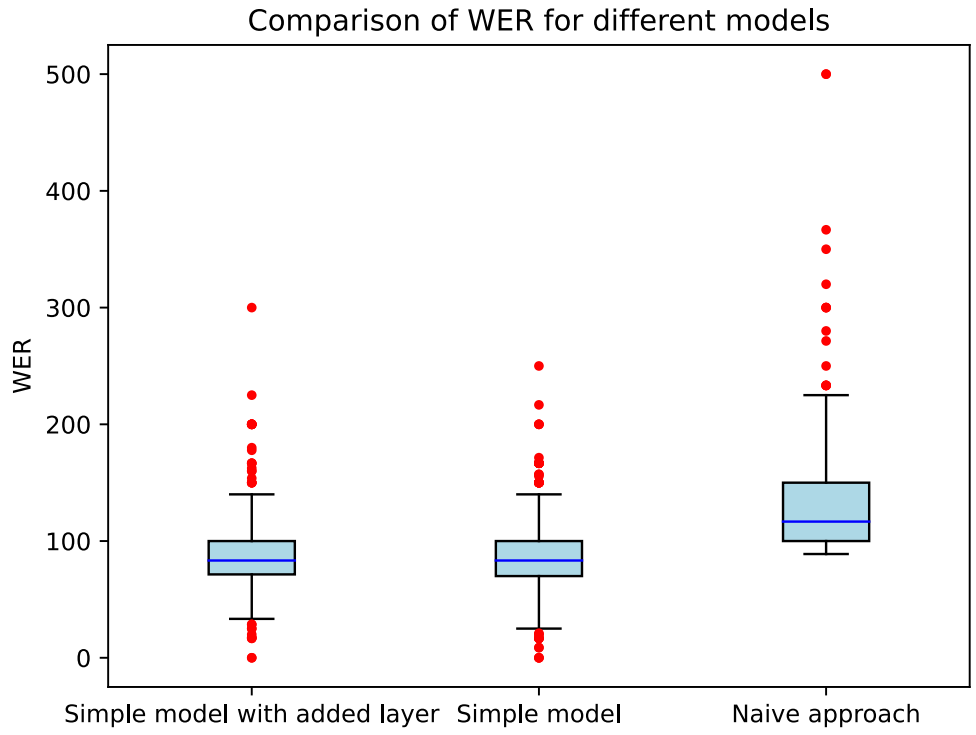

Figure 7.4 Comparison of WER [%] of different models. The compared models are a simple model with added linear layer, simple model with replaced layer and a model using naive approach.

Both simple models have similar distributions, suggesting they perform similarly. The naive approach shows a significantly worse performance and many more outliers, extending up to $WER = 500\%$.

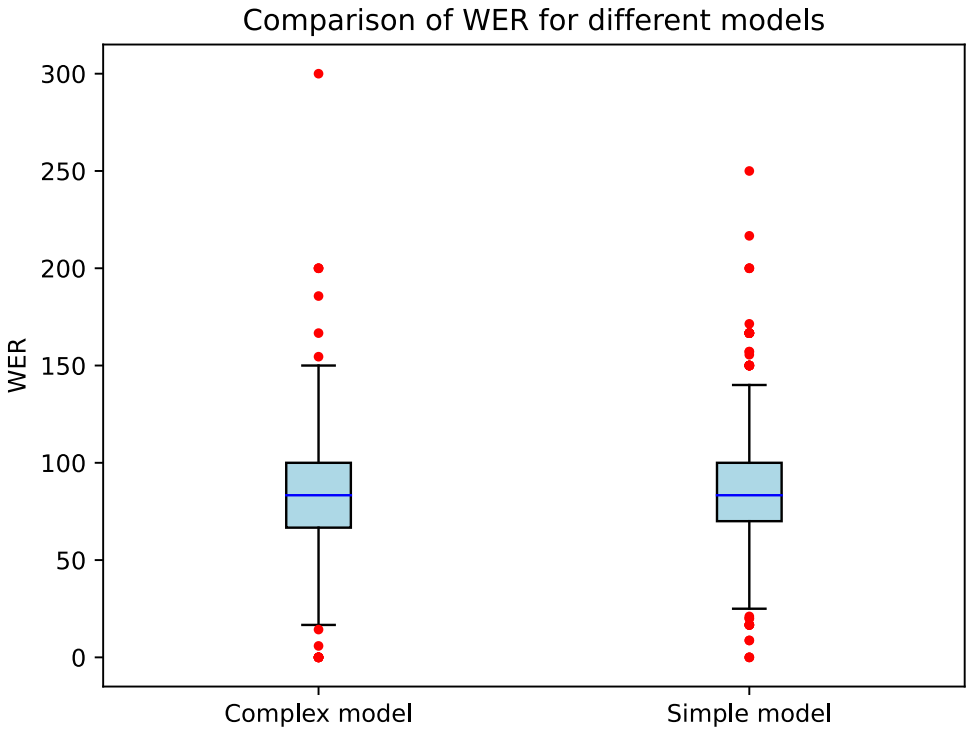

Figure 7.5 Comparison of WER [%] of the complex and the simple model trained with diacritics.

Figure 7.5 demonstrates that both models perform similarly on average, but the complex model shows a bit more variability in its results. The presence of outliers in both cases suggests that there are instances where both models can perform significantly better or worse than their typical performance.

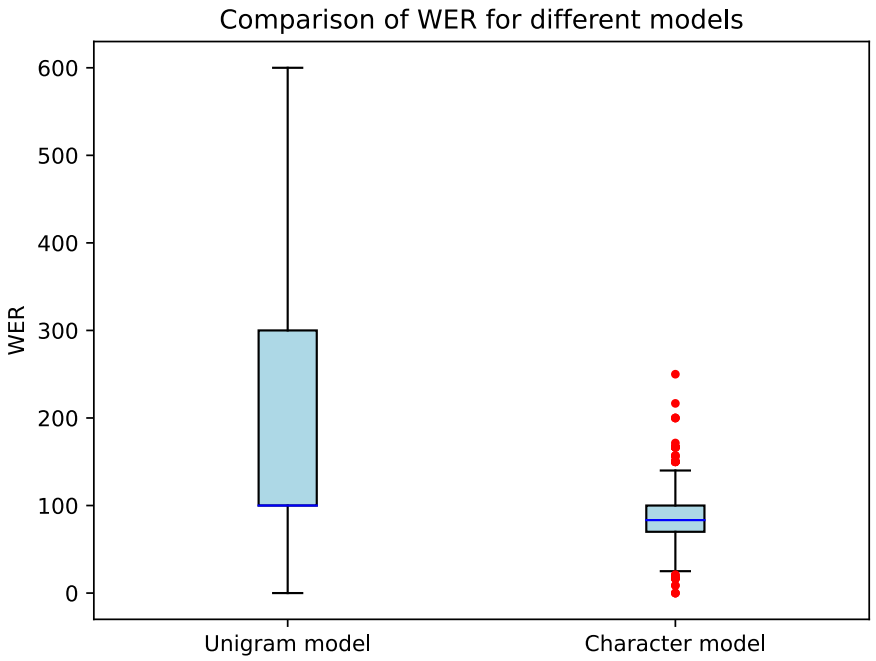

Figure 7.6 Comparison of WER [%] of unigram and character model with diacritics.

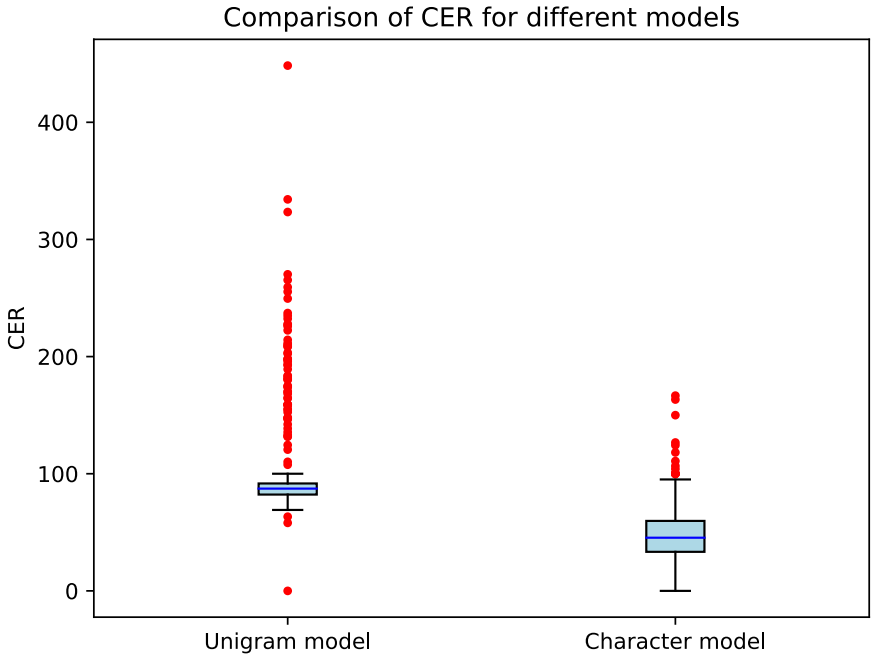

Figure 7.7 Comparison of CER [%] of unigram and character model with diacritics.

Chapter 8
Conclusion

In this thesis, I created the first lip reading model on Czech data under real-world conditions. This model performs sentence-level visual speech recognition on Czech videos in an unconstrained environment. It is able to detect speech in the Czech language with a word error rate of 84.8%. This currently does not allow full production deployment, but in my opinion, in some cases, it allows to capture the meaning of the conversation.

Training a lip reading model specifically for the Czech language presented unique challenges, primarily due to the scarcity of available Czech lip reading datasets. To overcome this limitation, I developed a specialized pipeline to facilitate the creation and processing of the necessary data. It has been demonstrated within this research that enlarging the dataset size notably enhances model performance. However, gathering, validating, and analyzing a large-scale dataset would be outside the scope of this thesis due to significant time constraints, even with the created pipeline. Instead, I utilized a smaller dataset that is sufficient to achieve meaningful conclusions within the allotted time frame.

8.1 Contribution

I did thorough research on the newest methods used in lip reading. I addressed the issue of the missing Czech data by developing a processing pipeline specifically for this task. To ensure the reliability of the data collection pipeline, I tested its correctness, paving the way for potentially large-scale data collection in future work. I collected a dataset from Czech TV shows, which involved negotiating usage and sharing permissions with Czech Television. Utilizing a state-of-the-art model applied to the Czech data, I explored various architectural improvements and training strategies to optimize the performance of the model.

8.2 Future work

Moving forward, future research should focus on expanding the dataset, as the findings strongly indicate that this will lead to significant improvements in the accuracy and overall effectiveness of the model. Future work should also focus on the development of an improved automatic pipeline for data collection. Enhancing the efficiency and robustness of the data collection process is critical, as it directly impacts the volume and quality of the data that can be acquired to train more accurate models. An optimized data collection pipeline could significantly reduce time and resource expenditure, making it feasible to collect large-scale data sets.

Real-time lip-reading systems would represent another milestone in this domain. Such systems would represent a breakthrough that would enable instantaneous speech recognition based

solely on visual cues. Although the first real-time AVSR models already exist, purely visual real-time speech recognition models have a long way to go. They cope with the strict response time interval for the procedure to be satisfactory. Even something like identifying ROI can take a significant amount of time, not to mention the recognition process itself.

Appendix A
Architectures

A.1 ResNet-18

Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connections
softmax	1000	

Figure A.1 Architecture of ResNet-18.

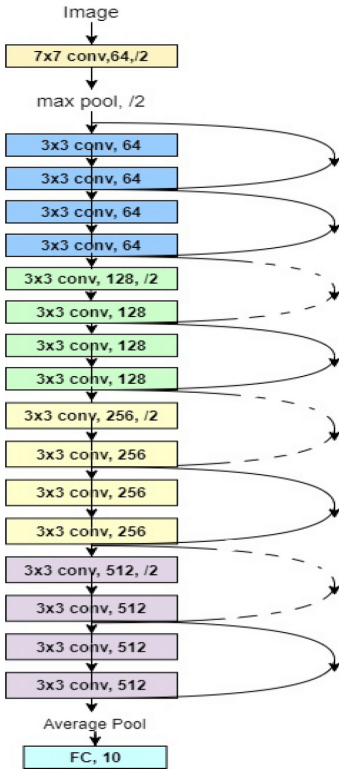

Figure A.2 Diagram of ResNet-18 architecture.

A.2 Front-end architecture

Component Name	Layer Type	Input Size	Output Size
Stem_1	Conv 3D, 5×7^2, 64	[B, 1, T_v, 88, 88]	[B, 64, T_v, 44, 44]
	3D Max Pooling, 1×3^2	[B, 64, T_v, 44, 44]	[B, 64, T_v, 22, 22]
Reshape	-	[B, 64, T_v, 22, 22]	[B×T_v, 64, 22, 22]
Residual Block_2	$\begin{bmatrix}\text{Conv 2D}, 3^2, 64\\ \text{Conv 2D}, 3^2, 64\end{bmatrix} \times 2$	[B×T_v, 64, 22, 22]	[B×T_v, 64, 22, 22]
Residual Block_3	$\begin{bmatrix}\text{Conv 2D}, 3^2, 128\\ \text{Conv 2D}, 3^2, 128\end{bmatrix} \times 2$	[B×T_v, 64, 22, 22]	[B×T_v, 128, 11, 11]
Residual Block_4	$\begin{bmatrix}\text{Conv 2D}, 3^2, 256\\ \text{Conv 2D}, 3^2, 256\end{bmatrix} \times 2$	[B×T_v, 128, 11, 11]	[B×T_v, 256, 6, 6]
Residual Block_5	$\begin{bmatrix}\text{Conv 2D}, 3^2, 512\\ \text{Conv 2D}, 3^2, 512\end{bmatrix} \times 2$	[B×T_v, 256, 6, 6]	[B×T_v, 512, 3, 3]
Aggregation	2D Global Average Pooling	[B×T_v, 512, 3, 3]	[B×T_v, 512, 1, 1]
Reshape	-	[B×T_v, 512, 1, 1]	[B, 512, T_v]

Figure A.3 "The architecture of the front-end encoder of the VSR model. The filter shapes are denoted by {Temporal Size × Spatial Size^2, Channels} and {Spatial Size^2, Channels} for 3D convolutional and 2D convolutional Layers , respectively. The sizes correspond to [Batch Size, Channels, Sequence Length, Height, Width] and [Batch Size × Sequence Length, Channels, Height, Width], for 3D and 2D convolutional layers, respectively. T_v denotes the number of input frames."
[21]

Appendix B
Acronyms

CNN Convolutional Neural Network

CTC Connectionist Temporal Classification

ReLU Rectified Linear Unit

LRW Lip reading words - dataset

LRS2 Lip reading sentences 2 - dataset

LRS3 Lip reading sentences 3 - dataset

Conformer - Convolution-augmented transformer

MHSA Multi-headed self-attention

FF Feed-forward

VSR Visual Speech Recognition

AVSR Audio-visual Speech Recognition

ROI Region of Interest

EOS End of Sequence

SOS Start of Sequence

Bibliography

1. PETAJAN, Eric David. *Automatic lipreading to enhance speech recognition (speech reading)*. University of Illinois at Urbana-Champaign, 1984.
2. EASTON, Randolph D; BASALA, Marylu. Perceptual dominance during lipreading. *Perception & Psychophysics*. 1982, vol. 32, pp. 562–570.
3. MCGURK, Harry; MACDONALD, John. Hearing lips and seeing voices. *Nature*. 1976, vol. 264, no. 5588, pp. 746–748.
4. KORTMANN, B. *English Linguistics: Essentials*. J.B. Metzler, 2020. ISBN 9783476056771. Available also from: `https://books.google.sk/books?id=0Bd4zQEACAAJ`.
5. RABINER, Lawrence R; JUANG, Biing-Hwang. *Fundamentals of Speech Recognition*. Philadelphia, PA: Prentice Hall, 1993. Prentice Hall signal processing series.
6. PALKOVÁ, Zdena. *Fonémy češtiny: Nový Encyklopedický Slovník češtiny*. 2017. Available also from: `https://www.czechency.org/slovnik/FON%C3%89MY%20%C4%8CE%C5%A0TINY`.
7. WOODWARD, Mary F; BARBER, Carroll G. Phoneme perception in lipreading. *Journal of Speech and Hearing Research*. 1960, vol. 3, no. 3, pp. 212–222.
8. FISHER, Cletus G. Confusions among visually perceived consonants. *Journal of speech and hearing research*. 1968, vol. 11, no. 4, pp. 796–804.
9. GOLDSCHEN, Alan J; GARCIA, Oscar N; PETAJAN, Eric D. Rationale for phoneme-viseme mapping and feature selection in visual speech recognition. In: *Speechreading by Humans and Machines: Models, Systems, and Applications*. Springer, 1996, pp. 505–515.
10. MOLL, Kenneth L; DANILOFF, Raymond G. Investigation of the timing of velar movements during speech. *The Journal of the Acoustical Society of America*. 1971, vol. 50, no. 2B, pp. 678–684.
11. BAHDANAU, Dzmitry; CHO, Kyunghyun; BENGIO, Yoshua. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*. 2014.
12. VASWANI, Ashish; SHAZEER, Noam; PARMAR, Niki; USZKOREIT, Jakob; JONES, Llion; GOMEZ, Aidan N; KAISER, Łukasz; POLOSUKHIN, Illia. Attention is all you need. *Advances in neural information processing systems*. 2017, vol. 30.
13. BELLO, Irwan; ZOPH, Barret; VASWANI, Ashish; SHLENS, Jonathon; LE, Quoc V. Attention augmented convolutional networks. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 3286–3295.
14. LU, Yiping; LI, Zhuohan; HE, Di; SUN, Zhiqing; DONG, Bin; QIN, Tao; WANG, Liwei; LIU, Tie-Yan. Understanding and improving transformer from a multi-particle dynamic system point of view. *arXiv preprint arXiv:1906.02762*. 2019.

15. GULATI, Anmol; QIN, James; CHIU, Chung-Cheng; PARMAR, Niki; ZHANG, Yu; YU, Jiahui; HAN, Wei; WANG, Shibo; ZHANG, Zhengdong; WU, Yonghui, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*. 2020.

16. JI, Shuiwang; XU, Wei; YANG, Ming; YU, Kai. 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*. 2012, vol. 35, no. 1, pp. 221–231.

17. KARPATHY, Andrej; TODERICI, George; SHETTY, Sanketh; LEUNG, Thomas; SUKTHANKAR, Rahul; FEI-FEI, Li. Large-scale Video Classification with Convolutional Neural Networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

18. FAN, Lei; XIA, Zhaoqiang; ZHANG, Xiaobiao; FENG, Xiaoyi. Lung nodule detection based on 3D convolutional neural networks. In: *2017 International conference on the frontiers and advances in data science (FADS)*. IEEE, 2017, pp. 7–10.

19. HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

20. FENGHOUR, Souheil; CHEN, Daqing; GUO, Kun; LI, Bo; XIAO, Perry. Deep learning-based automated lip-reading: A survey. *IEEE Access*. 2021, vol. 9, pp. 121184–121205.

21. MA, Pingchuan; HALIASSOS, Alexandros; FERNANDEZ-LOPEZ, Adriana; CHEN, Honglie; PETRIDIS, Stavros; PANTIC, Maja. Auto-AVSR: Audio-visual speech recognition with automatic labels. In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

22. JEFFERS, J.; BARLEY, M. *Speechreading (lipreading)*. Thomas, 1971. ISBN 9780398021856. Available also from: `https://books.google.sk/books?id=-UNlGOqM5eUC`.

23. NETI, Chalapathy; POTAMIANOS, Gerasimos; LUETTIN, Juergen; MATTHEWS, Iain; GLOTIN, Herve; VERGYRI, Dimitra; SISON, June; MASHARI, Azad. Audio visual speech recognition. 2000.

24. BOZKURT, Elif; ERDEM, Cigdem Eroglu; ERZIN, Engin; ERDEM, Tanju; OZKAN, Mehmet. Comparison of phoneme and viseme based acoustic units for speech driven realistic lip animation. In: *2007 3DTV Conference*. IEEE, 2007, pp. 1–4.

25. ZHAO, Guoying; PIETIKÄINEN, Matti; HADID, Abdenour. Local spatiotemporal descriptors for visual recognition of spoken phrases. In: *Proceedings of the international workshop on Human-centered multimedia*. 2007, pp. 57–66.

26. GRAVES, Alex; MOHAMED, Abdel-rahman; HINTON, Geoffrey. Speech recognition with deep recurrent neural networks. In: *2013 IEEE international conference on acoustics, speech and signal processing*. Ieee, 2013, pp. 6645–6649.

27. MA, Pingchuan; PETRIDIS, Stavros; PANTIC, Maja. Visual speech recognition for multiple languages in the wild. *Nature Machine Intelligence*. 2022, vol. 4, no. 11, pp. 930–939.

28. NODA, Kuniaki; YAMAGUCHI, Yuki; NAKADAI, Kazuhiro; OKUNO, Hiroshi G; OGATA, Tetsuya, et al. Lipreading using convolutional neural network. In: *Interspeech*. 2014, vol. 1, p. 3.

29. GARG, Amit; NOYOLA, Jonathan; BAGADIA, Sameep. Lip reading using CNN and LSTM. *Technical report, Stanford University, CS231 n project report*. 2016.

30. LI, Yiting; TAKASHIMA, Yuki; TAKIGUCHI, Tetsuya; ARIKI, Yasuo. Lip reading using a dynamic feature of lip images and convolutional neural networks. In: *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*. IEEE, 2016, pp. 1–6.

31. FUNG, Ivan; MAK, Brian. End-to-end low-resource lip-reading with maxout CNN and LSTM. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2511–2515.
32. WENG, Xinshuo; KITANI, Kris. Learning spatio-temporal features with two-stream deep 3d cnns for lipreading. *arXiv preprint arXiv:1905.02540*. 2019.
33. FENG, Dalu; YANG, Shuang; SHAN, Shiguang. An efficient software for building LIP reading models without pains. In: *2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2021, pp. 1–2.
34. YANG, Shuang; ZHANG, Yuanhang; FENG, Dalu; YANG, Mingmin; WANG, Chenhao; XIAO, Jingyun; LONG, Keyu; SHAN, Shiguang; CHEN, Xilin. LRW-1000: A naturally-distributed large-scale benchmark for lip reading in the wild. In: *2019 14th IEEE international conference on automatic face & gesture recognition (FG 2019)*. IEEE, 2019, pp. 1–8.
35. MARGAM, Dilip Kumar; ARALIKATTI, Rohith; SHARMA, Tanay; THANDA, Abhinav; ROY, Sharad; VENKATESAN, Shankar M, et al. LipReading with 3D-2D-CNN BLSTM-HMM and word-CTC models. *arXiv preprint arXiv:1906.12170*. 2019.
36. SAITOH, Takeshi; ZHOU, Ziheng; ZHAO, Guoying; PIETIKÄINEN, Matti. Concatenated frame image based CNN for visual speech recognition. In: *Computer Vision–ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part II 13*. Springer, 2017, pp. 277–289.
37. CHUNG, Joon Son; ZISSERMAN, Andrew. Lip reading in the wild. In: *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part II 13*. Springer, 2017, pp. 87–103.
38. ASSAEL, Yannis M; SHILLINGFORD, Brendan; WHITESON, Shimon; DE FREITAS, Nando. Lipnet: End-to-end sentence-level lipreading. *arXiv preprint arXiv:1611.01599*. 2016.
39. STAFYLAKIS, Themos; TZIMIROPOULOS, Georgios. Combining residual networks with LSTMs for lipreading. *arXiv preprint arXiv:1703.04105*. 2017.
40. PETRIDIS, Stavros; STAFYLAKIS, Themos; MA, Pingehuan; CAI, Feipeng; TZIMIROPOULOS, Georgios; PANTIC, Maja. End-to-end audiovisual speech recognition. In: *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 6548–6552.
41. AFOURAS, Triantafyllos; CHUNG, Joon Son; ZISSERMAN, Andrew. Deep lip reading: a comparison of models and an online application. *arXiv preprint arXiv:1806.06053*. 2018.
42. MA, Pingchuan; PETRIDIS, Stavros; PANTIC, Maja. End-to-end audio-visual speech recognition with conformers. In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7613–7617.
43. SHILLINGFORD, Brendan; ASSAEL, Yannis; HOFFMAN, Matthew W; PAINE, Thomas; HUGHES, Cían; PRABHU, Utsav; LIAO, Hank; SAK, Hasim; RAO, Kanishka; BENNETT, Lorrayne, et al. Large-scale visual speech recognition. *arXiv preprint arXiv:1807.05162*. 2018.
44. NODA, Kuniaki; YAMAGUCHI, Yuki; NAKADAI, Kazuhiro; OKUNO, Hiroshi G; OGATA, Tetsuya. Lipreading using convolutional neural network. In: *fifteenth annual conference of the international speech communication association*. 2014.
45. FENGHOUR, Souheil; CHEN, Daqing; GUO, Kun; XIAO, Perry. Lip reading sentences using deep learning with only visual cues. *IEEE Access*. 2020, vol. 8, pp. 215516–215530.
46. KOLLER, Oscar; NEY, Hermann; BOWDEN, Richard. Deep learning of mouth shapes for sign language. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2015, pp. 85–91.

47. SON CHUNG, Joon; SENIOR, Andrew; VINYALS, Oriol; ZISSERMAN, Andrew. Lip reading sentences in the wild. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 6447–6456.

48. TAMURA, Satoshi; NINOMIYA, Hiroshi; KITAOKA, Norihide; OSUGA, Shin; IRIBE, Yurie; TAKEDA, Kazuya; HAYAMIZU, Satoru. Audio-visual speech recognition using deep bottleneck features and high-performance lipreading. In: *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2015, pp. 575–582.

49. PETRIDIS, Stavros; PANTIC, Maja. Deep complementary bottleneck features for visual speech recognition. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 2304–2308.

50. YU, Jianwei; ZHANG, Shi-Xiong; WU, Jian; GHORBANI, Shahram; WU, Bo; KANG, Shiyin; LIU, Shansong; LIU, Xunying; MENG, H.; YU, Dong. Audio-Visual Recognition of Overlapped Speech for the LRS2 Dataset. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020, pp. 6984–6988. Available from DOI: 10.1109/ICASSP40776.2020.9054127.

51. AFOURAS, Triantafyllos; CHUNG, Joon Son; ZISSERMAN, Andrew. LRS3-TED: a large-scale dataset for visual speech recognition. *arXiv preprint arXiv:1809.00496*. 2018.

52. TROJANOVÁ, Jana; HRÚZ, Marek; CAMPR, Pavel; ŽELEZNỲ, Miloš. Design and recording of czech audio-visual database with impaired conditions for continuous speech recognition. 2008.

53. SHRIBERG, Elizabeth; STOLCKE, Andreas; HAKKANI-TÜR, Dilek; TÜR, Gökhan. Prosody-based automatic segmentation of speech into sentences and topics. *Speech communication*. 2000, vol. 32, no. 1-2, pp. 127–154.

54. DENG, Jiankang; GUO, Jia; ZHOU, Yuxiang; YU, Jinke; KOTSIA, Irene; ZAFEIRIOU, Stefanos. Retinaface: Single-stage dense face localisation in the wild. *arXiv preprint arXiv:1905.00641*. 2019.

55. BULAT, Adrian; TZIMIROPOULOS, Georgios. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 1021–1030.

56. DAUPHIN, Yann N; FAN, Angela; AULI, Michael; GRANGIER, David. Language modeling with gated convolutional networks. In: *International conference on machine learning*. PMLR, 2017, pp. 933–941.

57. GRAVES, Alex; FERNÁNDEZ, Santiago; GOMEZ, Faustino; SCHMIDHUBER, Jürgen. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 369–376.

58. WATANABE, Shinji; HORI, Takaaki; KIM, Suyoun; HERSHEY, John R; HAYASHI, Tomoki. Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*. 2017, vol. 11, no. 8, pp. 1240–1253.

59. LOSHCHILOV, Ilya; HUTTER, Frank. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*. 2017.

60. KUDO, Taku. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*. 2018.

Content of the attachment

- `src` ..... the directory with source codes
- `thesis.pdf` ..... the thesis text in PDF format
- `results.zip` ..... zip file with results of experiments in csv files
- `models.zip` ..... zip file containing different models