Danmarks Tekniske Universitet

Fagprojekt 02466

Deep Fake Voice-Conversion and Ethical Considerations of such technology

August Semrau Andersen - s183918
Karl Byberg Ulbæk - s183931
William Diedrichsen Marstrand - s183921

June 24, 2020, Danmarks Tekniske Universitet

Abstract

This project investigates, to which degree voice conversion technologies can be utilized for transforming dialect heavy speech into a standard voice, to improve the performance of the existing Danish state-of-the-art speech to text system danspeech.

The two voice conversion models, StarGAN and Variational Autoencoder, were implemented for this task as they both on paper were capable of converting from any speaker not already included in training of their model. StarGAN overall showed a very convincing performance, producing good sounding conversions that matched the target speaker closely. The implemented version of Variational Autoencoder produced unconvincing conversions, preserving little to no linguistic information and lagging transformation of speaker styles.

The transcriptive performance of danspeech, measured by word-error-rate, was tested on original speech samples, StarGAN converted speech samples and Variational Autoencoder converted speech samples. The transcriptive performance was found to be best using original speech and slightly worse using StarGAN converted speech, while danspeech was almost unable to transcribe Variational Autoencoder converted speech samples.

The overall proposed idea with implementing voice conversion was that a speech-to-text danspeech model retrained on audio converted to a single speaker would be better at transcribing converted speech than a conventionally trained danspeech model transcribing original speech. Danspeech was retrained using the same data set in original voices and in the StarGAN converted voices. The word-error-rate performances of the two differently retrained danspeech models showed that danspeech trained on original speech performed better than the one trained on converted data, and so the proposed setup was not found to be advantageous yet still holding potential for further improvement.

The paper discusses these results and argues that investigating improvements especially in audio-quality of converted speech could potentially help make the proposed setup viable. Lastly the ethical considerations related to utilizing voice conversion technology are discussed with a focus on Safe AI principles and the importance of equal access to speech services independent of dialect.

Contents

Acronyms

VC - Voice Conversion

STT - Speech To Text

GAN - Generative Adversarial Network

VAE - Variational Autoencoders

WER - Word Error Rate

Glossary

VC model - Is a model that converts one speakers voice to another speakers voice.

STT model - Is a model that transcribes spoken words into written words.

Acoustic features - The sounds of speech that can be recorded and captured in wave forms e.g. frequency and amplitude.

Speaker Style - Acoustic features defining a particular speaker.

Linguistic features - The meaning or information contained in the words uttered by a speaker.

Dialect - A language variation characterizing a group of speakers within the same language. This variation is characterized by slight changes in the acoustic features of an utterance while still maintaining the same linguistic content.

1 Introduction

All code for the project is publicly available at https://gitlab.gbar.dtu.dk/s183921/fagprojekt2020

1.1 Motivation and Purpose

Spoken language is constantly changing creating variations within the language, known as dialects [22]. This is certainly the case in Denmark, where studies have proved changes within the Danish language throughout the past decade till present day [7]. In this project groups of people speaking a different dialect than the standard of a country, for which data is also scarce, are defined as dialect minorities. The status of dialect minority directly affects these groups, as the challenge and high cost associated with gathering representative speech data leads to a lack of representative speech corpora, inhibiting their use of different speech technologies [12].
In Denmark the danspeech [25] initiative is one of the most comprehensive Danish speech to text (STT) frameworks, but they too perform suboptimally on non-standard dialect speech due to the lack of data from certain Danish dialect minorities. This proposes a challenge in which danspeech are dependent on the improvement of diversity in their utilized speech corpora, but the cost of gathering such data simply is too high.

Different projects, such as Mozilla Common Voice [34], have been initiated to support the creation of publicly available open source speech corpora for different languages. However, data is scarce for lesser spoken languages like Danish, where no usable corpora exists at the moment. Also, at least for now, the focus of Common Voice is at an overall language level, leaving out dialect annotations/focus.

This project explores another option, with the aim of eliminating the need for large speech corpora on different dialects and instead use Voice Conversion (VC) technologies such as StarGAN-VC [13] and Variational Autoencoder [23]. With VC, a model converts speech from one or multiple source speakers to speech from one or multiple target speakers, making the source speaker sound like the target speaker. The use case is then to convert Danish dialect minorities, source speakers, into a common dialect such as *standard Danish*, target speaker. With this approach, only large amounts of data are needed for the target speaker voice, while few or no data samples of dialect minority source speakers would be required. Utilizing voice conversion could potentially improve Danish speech technologies, as they could be trained on the standard target voice alone, for which larger amounts of training data would be accessible.

1.2 Technical Context of VC

In regards to state-of-the-art voice conversion models, there are three key factors in determining which fits the needs of this project.
Firstly, VC models can work on different types of data, parallel and non-parallel. The models utilized in this project use non-parallel data, in which the utterances of source and target speakers do not need to be the same and do not need to be time aligned. This is opposed to parallel data where speakers need to make the exact same utterances and the utterances should be time aligned [18].

Secondly, it is important to consider the type of voice conversion framework to be used. Generally three types are available: many-to-many, many-to-one, or one-to-many. The first meaning voice conversion from an arbitrary speaker to another arbitrary speaker, the second being from an arbitrary to a specific speaker, and the last a specific to an arbitrary speaker [2]. There is some debate about naming conventions as some also like to use *any* instead of *many* [14]. For this project the *any* term will be used for describing mapping to or from unseen speakers, as the term clearly describes the arbitrary aspect, that *any* speaker can be used. Furthermore only any-to-any and any-to-one models will be considered here, as only these two are fit for the problem of mapping multiple dialect speakers to a single standard speaker voice.
Adding to this, the models need capabilities in performing zero-shot conversions, in which the source and/or target speaker have not been involved in the training of the model. The models are therefore able to convert from before-unseen source speakers, as retraining the model for each new source speaker would be unfeasible.

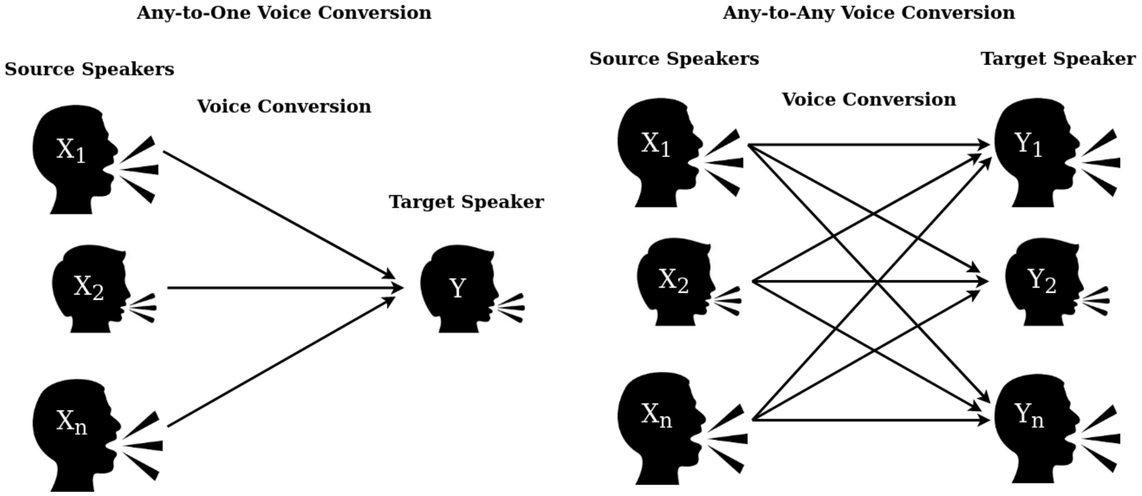


Figure 1: Any-to-One and Any-to-Any Voice Conversion

Lastly, the feasibility of training the models needs consideration as well, and in the context of this project, particularly two criteria are important.
First, the amount of source speaker training data needed is crucial. As explained earlier, dialect minority speech data is scarce and expensive to collect, which creates the need for models that can

perform satisfying conversions on few or close to no dialect minority speech samples.
Second, the time and hardware needed to train the models need to align with available resources. For this reason, WaveNet [10] by DeepMind [43] has been ruled out, even though it has shown promising results as the new state-of-the-art VC model today.

1.3 Problem Statement

This project specifically looks at, to which degree voice conversion technologies can be utilized for transforming dialect heavy speech into a standard voice, to improve the performance of the existing Danish state-of-the-art speech to text system danspeech.

1.3.1 Research Questions

To guide the investigation a set of research questions have been formulated.

- How well can state-of-the-art voice conversion results from StarGAN-VC[13] and Instance-normalization[17] VC models be reproduced for many-to-one, zero-shot voice conversion scenarios?
- How does the danspeech speech to text translation perform when applying voice conversion models compared to using no voice conversion?
- How does the danspeech speech to text translation perform when voice converted input is provided to a pretrained danspeech model compared to a danspeech model retrained on voice converted data?

1.4 Project Overview

This project initially presents the data utilized in investigating the research questions under section 2, as it lays the foundation for the implementations of VC models. The overall approach to how the problem statement was investigated is explained under section 3.1. The set of applied voice conversion models are then presented in sections 3.2 and 3.3, and related to the use case of converting different dialects to a single standard voice. This is followed closely by detailed descriptions of their actual implementation, sections 3.4 and 3.5 respectively. The utilized speech to text framework is presented in section 3.6, followed by section 3.7 on evaluating the results the three research questions investigating the implementation of voice conversion in STT technologies. Finally the results of the three research questions are presented under sections 4.1, 4.2 and 4.3 and these are discussed under section 5.1. Here, ideas for potential future investigation and technical improvements are discussed, see section 5.2 for that, as well as ethical considerations that should be taken into account, section 5.3.

2 Data

This paper will make use of a total of 4 different data sets. The reasoning behind choosing each of the 4 data sets will be discussed in the following sections together with their characteristics, features and shortcomings.

2.1 CSTR VCTK Corpus

The Center for Speech Technology, Voice Cloning Toolkit, from here on the VCTK corpus, is a publicly available data set created by The University of Edinburgh [24]. The data has been used by Google Deep-mind in the first implementation of WaveNet [43], and in many other projects. The data set contains speech by 109 individual native English speakers, with various accents, each reading out around 400 sentences recorded as individual 10 second long .wav files.
The data set is of high quality, with a negligible amount of noise and loud and clear speaker pronunciations. The VCTK data set was used in both the original StarGAN and VAE papers to train their respective models, making it the obvious choice for initial pilot experiments with the purpose of getting the VC-models up and running. The English VCTK data was of no further interest as this project exclusively is interested in Danish voice conversion.

2.2 Spraakbankens Danish Data Set

Accomplishing VC in a Danish-spoken setting requires Danish voice data. The data that best suited this project's needs was obtained from a database by the name of Spraakbanken, made available by the Norwegian National library [36].
Spraakbanken provides a wide variety of data sets in multiple different languages, but most importantly offers an extensive Danish data set in terms of quantity and diversity. The Danish data set from Spraakbanken will be denoted Spraakbanken or Spraakbanken corpus from here on and is comprised of 2 parts:
A *training* set which consists of 614 individual speakers each with 312 voice clips, with a few exception with less clips.
A *test* set of 55 different speakers with 987 voice clips per speaker.
The Spraakbanken naming convention of *train* and *test* set will be continued in this paper. Both data sets combined amounts to a total of 63 gigabyte speaker data. All data is in 16 kHz, 16 bit stereo audio. The *test* and *training* data set will each serve their own purpose but for now their common characteristics will be discussed together. Spraabankens Danish data set offers a wide variety of compelling features in favor of this papers objective.
First of, the data set is very well documented in numerous ways, which will benefit this paper. The data is fully transcribed. For every voice clip there exist a text file containing the corresponding linguistic content, which is essential when retraining danspeech described later under section 3.6. Each speaker has their age and gender recorded. Lastly the speakers have their geographical origin noted as well. This fact is fairly important as geographical origin roughly translates into which dialect the speakers possess, which ties directly into the primary goal of this paper. As briefly mentioned, the quantity and especially diversity of the data is impressive.

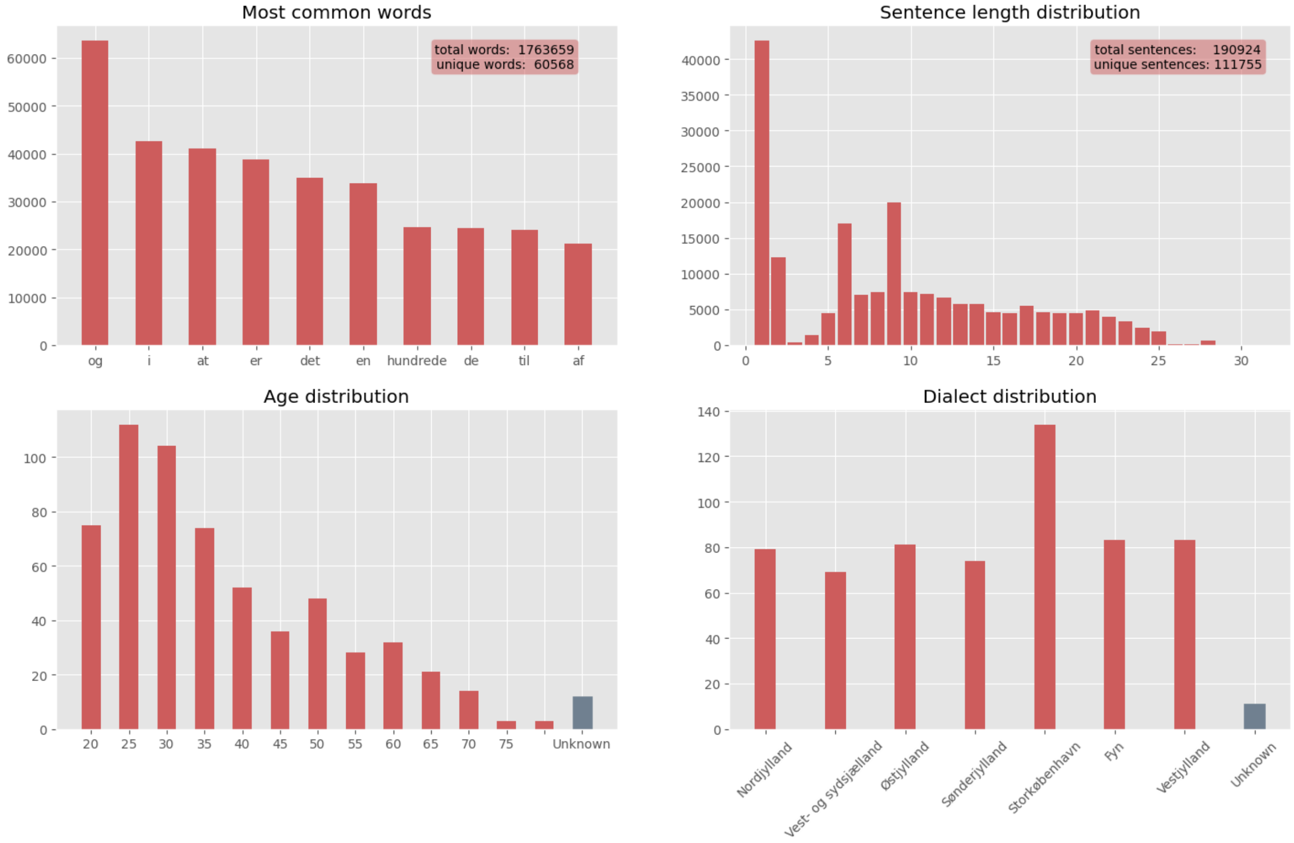


Figure 2: Plot displaying most common words, distribution of sentence lengths, age distribution and dialect distribution of the full Spraakbanken *training* data set containing 614 speakers with 312 voice clips each.

The diversity is clearly underlined by inspecting the amount of unique sentences. 111755 unique sentences to be exact, which corresponds to more than half of the utterances. It is often common that voice data sets have speakers utter the exact same sentences, but this is clearly not the case with Spraakbanken. The diversity claim is further supported by the amount of unique individual words, though it should be noted that words were not stemmed in the plot meaning that same words with different suffixes will be counted separately. The data set having this big of a combined vocabulary is less important for training the VC models, but vital for retraining danspeech.
The data set is clearly has an abundance of one word sentences, which is not optimal, but the rest of the sentence lengths are somewhat evenly represented.

From the age and dialect distributions, diversity again becomes the keyword with the exception of dialects being slightly dominated by Storkøbenhavn, and the age distribution being slightly under represented by older people. On the subject of different dialects: the plotted distribution are based on what labels the speakers had been given by Spraakbanken and only serve as an indicator. In general, the dialects were for the most part not very prominent. The gender distribution is almost 50/50 and can be found in the appendix C.

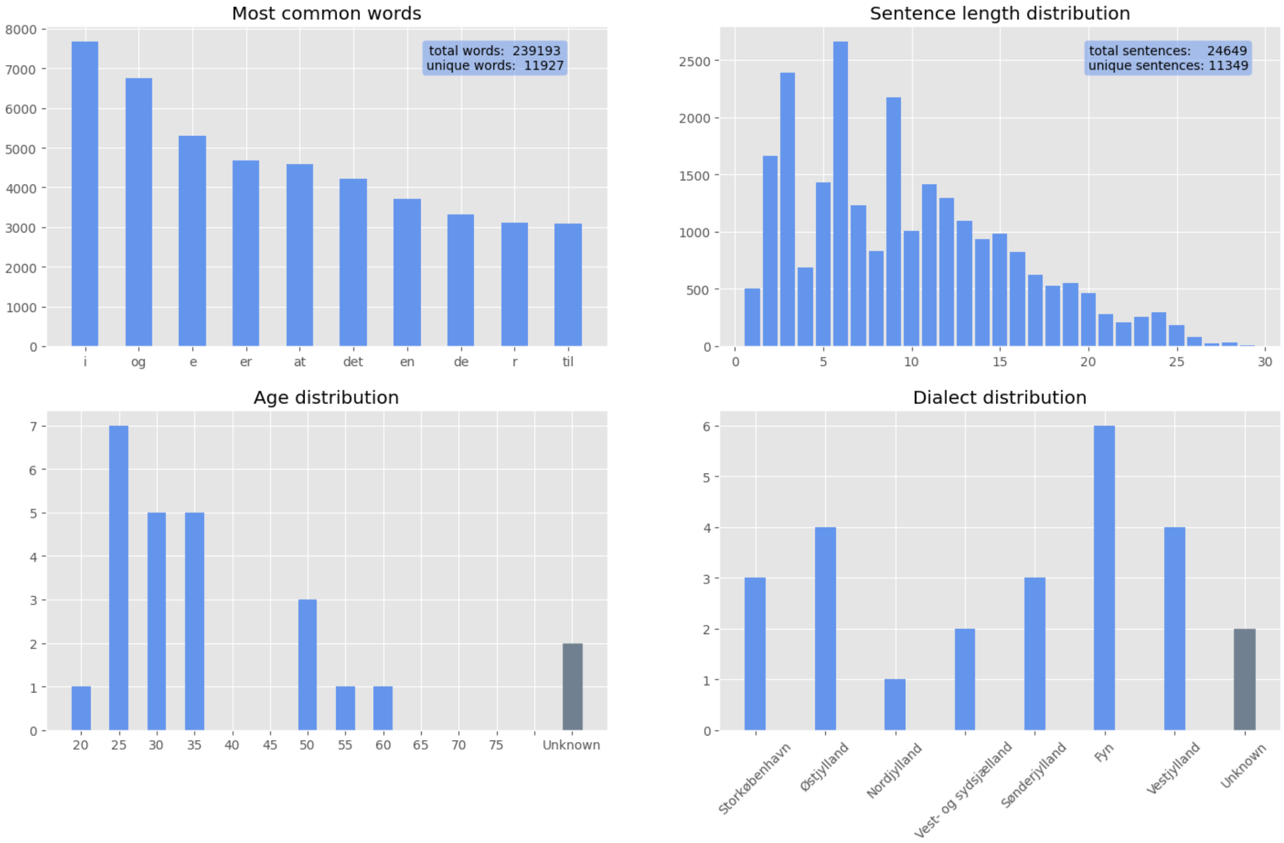


Figure 3: Plot of the 25 of 55 speakers from Spraakbanken *test* data set, each with 987 voice clips.

Only the 25 of the total 55 speakers of the Spraakbanken *test* set were plotted, due to only these being used in the actual retraining of StarGAN and VAE. For the most part, the observations and findings about the Spraakbanken *train* data also hold true for the *test* data. The main differences being that the *test* data is smaller in size given its fewer speakers although it has more utterances per speaker. The *test* data does not have an abundance of 1 word sentences, but is slightly over represented with regards to speakers from Fyn.

Upon extraction of the transcripts (from the format they were given by Spraakbanken to something more useable), an automated random check was performed. The random check consisted of 20 samples per speaker and was performed to ensure that the extraction had been performed correctly and more importantly that the transcripts in fact matched the audio files. Upon this random check it was discovered that some of the audio data was faulty in the sense that out-takes/retakes had been left in by accident. It is difficult to state the magnitude of these faults as the random check only consisted of 20 samples per speaker. About 20 instances across 10 speakers were caught this way and the 3 speakers having 3+ faults were removed entirely from the dataset.

In general, the overall quality of the voice data from Spraakbanken was suboptimal. The data was often marked by an underlying constant electronic noise, and the voices were not always crisp and clear by today's standard. The recordings are from the start of this century and even in that context their equipment seems subpar by the sound of it.

As stated the Spraakbanken corpus will serve 2 purposes. The full *training* set will be used for retraining danspeech due to its sheer magnitude and diversity, and as previously mentioned parts of the *test* set will be used for training the VC models.

2.3 Evaluation Data

The data set used for evaluating the success of the VCs by means of the Word Error Rate (described under section 3.7.3), and McNemar test (described under section 3.7.4), has been fully built from the ground up. There are numerous reason for this:

It is not apparent from the danspeech documentation exactly what data their primary model is trained on. As it is extremely important not to test on trained data, none of Spraakbanken can be used for evaluation. If danspeech is given its own training data as input it will translate it 100% accurately, making the impact of the VCs insignificant and undetectable. Furthermore, the speakers dialect labels as assigned by Spraakbanken are only indicators of their dialects and should by no means be considered definite. In order to fully test the capabilities and the impact of the VC models as a tool to standardize dialect heavy speech, the models should be faced with proper and representative, dialect heavy speech. To ensure this criteria was met it was decided that each voice clip of the evaluation set had to be handpicked. The features of the evaluation data are as follows:

- The size of the set is three male and three female speakers per region.
- The regions in question were limited to Nord- og midtjylland, Fyn, Syd- og vestsjælland, Storkøbenhavn og Sønderjylland. The limited number of regions, was due to complications with finding suitable audio for some regions.
- All speakers are unique and all spoken sentences are unique.
- Each speaker has one rather long voice clip, though some of the voice clips are made up of multiple individual clips by the same speaker in the same setting. This was necessary to get enough continuous data on some speakers.
- The clips are on average 41 words and all clips were handpicked from YouTube videos.
- The quality of the data is generally comparable to that of the Spraakbanken data, although the quality varies more between clips.

2.4 Audio Data-Characteristics

When being utilized in the two VC-models, the acoustic features of above described audio data is turned into utilizable representations on which training and conversion can take place. This is described later under StarGAN section 3.2 and VAE 3.3.
Before this though, audio data can be graphed and understood as an acoustic waveform representing variations in air pressure over time, the definition of sound. This essentially means that what distinguishes each speaker in the dataset is a sum of unique acoustic features derived from the waveform their speech produces. These unique speaker features are what the VC-models use in order to generate the speakers voice.

The data used in this project is 16-bit audio sampled in 16 kHz.
16-bit audio is considered as the standard bit depth and relates to the resolution with which the audio is recorded.
The sampling frequency of the data, measured as the number of sampled data-points per second (Hertz), determines what acoustic frequencies can be included in an audio recording. From the Nyquist Theorem it is described that the highest frequency, that can be represented accurately is one half of the sampling rate. The human ear is able to hear sound in the frequency range of 20 Hz to 20,000 Hz, so in order to record audio that can represent the entire hearable spectrum, two samples are needed for every frequency. This results in having a desirable sampling rate that is at least 40 kHz, the general standard sampling rate actually being higher, around 48 kHz. In the context of voice conversion and STT, the high sampling rates are not necessarily required as human voice only ranges from 300-3000hz. Combine this with the fact that higher sampling rate results in greater amounts of data to process, all systems relating to audio data used in this project uses downsampled speech data with a 16 kHz sampling rate.

3 Method

3.1 Overall Approach

The overall idea proposed by this project consists in utilizing voice conversion as a means of standardizing input data for a speech-to-text model. This project will investigate whether the implementation of voice conversion into a conventional speech-to-text setup can in fact improve the performance of said STT setup.

3.1.1 Conventional Speech-to-Text Setup

The conventional STT setup consists of two separate tasks which are illustrated in figure 4 below. The first task, illustrated in the top of the figure, is the training of an STT model which generally takes large amounts of training data consisting of different speaker-utterance-samples and their written equivalents (labels). It essentially trains itself to become a fully trained STT model that is able recognize which words are spoken and transcribe them into their written form.
The second task, illustrated bottom of figure 4, is the actual transcription performed by the fully trained STT model. Here it is given one or more utterances not included in training and outputs their transcription.

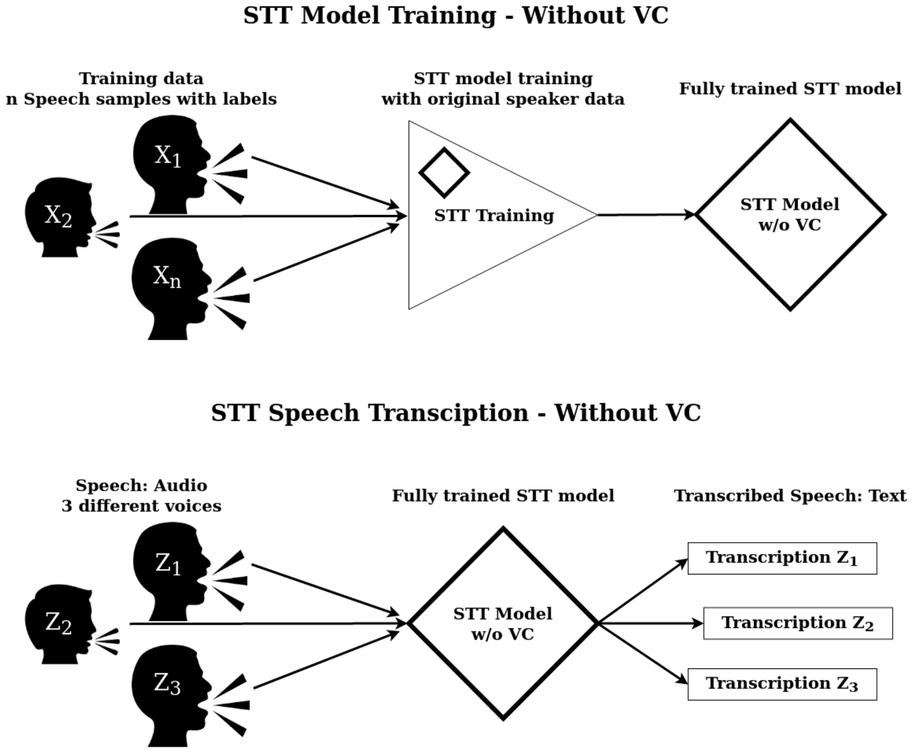


Figure 4: Conventional STT setup. Top: Training task to form fully trained STT model. Bottom: Transcription task using fully trained STT model.

3.1.2 Proposed VC-Implemented Speech-to-Text Setup

Described below is the new proposed setup, illustrated in figure 5, which adds another task and introduces VC to the conventional STT setup.

First task of the proposed setup, illustrated top in figure 5, consist in training a voice conversion model to perform any-to-one conversions. This VC training is done on a large number speech samples from several unique speakers who represent a diverse spectrum of voices.
The second step, illustrated middle, is the new training setup for the STT model. Before training of the model can take place, the large number of different voices and their labeled utterances are converted into a common voice. This means that while the amount of training data for the STT remains the same, the spectrum of different voices it has to recognize is reduced to merely one.
The third task, illustrated bottom, is the transcription of utterances using VC. The fully VC-trained STT model will now need the input speech to be of the same neutral common voice that it was trained on, as to why all new utterances are first converted into this very voice. After conversion, these are inputted into the STT model and outputted as transcriptions.

The idea is that, as the STT model only needs to be able to transcribe one single voice, and the amount of training data for this voice is hugely increased, it will perform better that the conventional setup allows for.

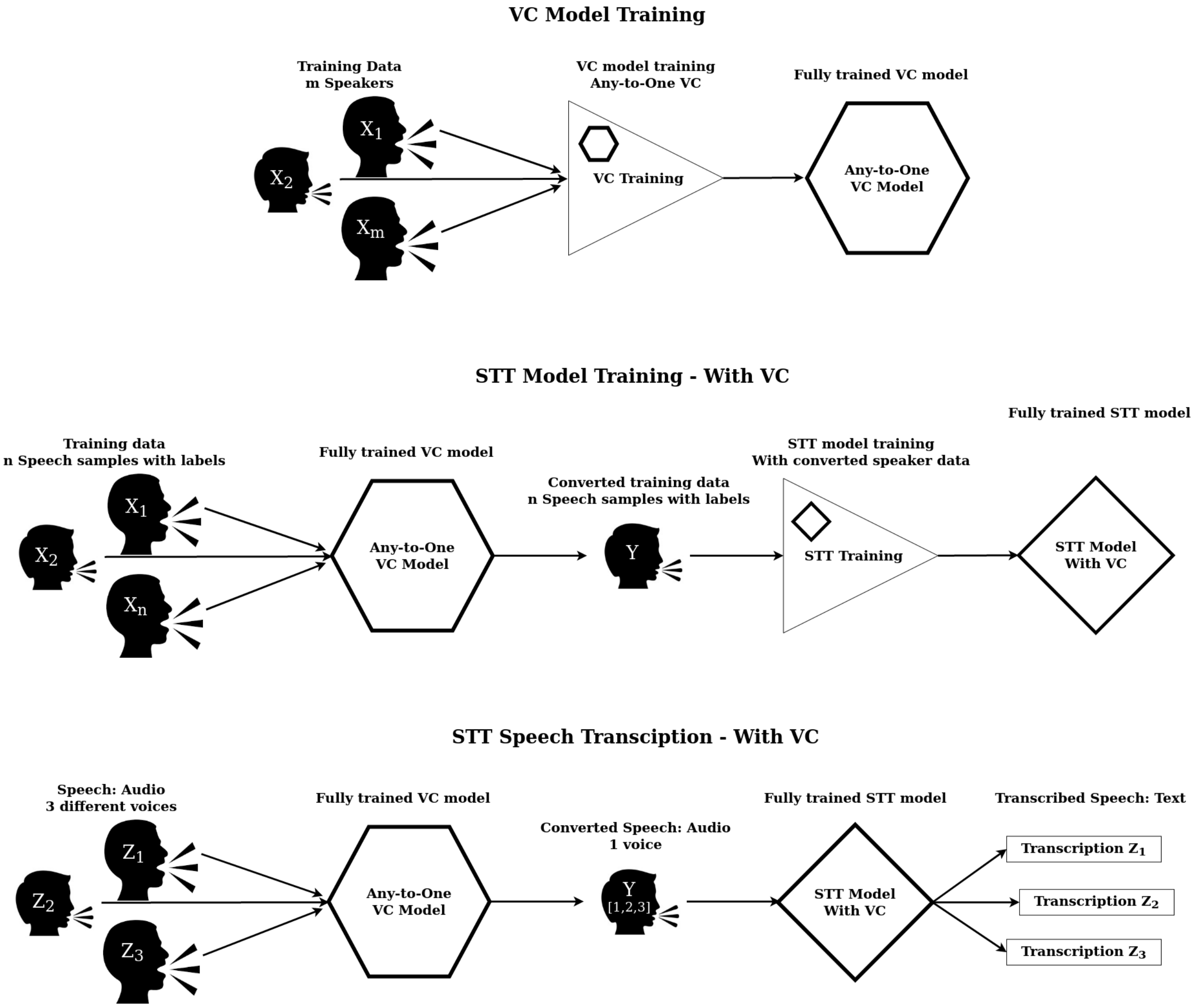


Figure 5: New STT setup incorporating VC technology. Top: Training of VC model on different speakers. Middle: Training of STT model using converted training data. Bottom: Transcription process on converted speech using trained STT model.

3.2 StarGAN-VC

The first of two state-of-the-art VC models used in this project is from 2018; 'StarGAN-VC: Non-parallel many-to-many voice conversion with Star Generative Adversarial Networks[13].
StarGAN-VC improves upon an earlier publication, CycleGAN-VC [11], but at the core of each of them is the Generative Adversarial Network(GAN).

3.2.1 Generative Adversarial Networks

A Generative Adversarial Network, just GAN from now on, first proposed by Ian J. Goodfellow [5], is in its most basic form a combination of two neural networks pitted against each other; a generative network and a discriminatory network. Based upon some initial input data distribution, this could be simple coordinates, natural images etc. or in the case of this project speech and its acoustic features, the generative network works to generate a set of model data from the data distribution. The discriminatory network works to classify whether a given sample is from the generated model distribution (fake) or the data distribution (real). The training objective of the generative model is to maximize the error rate of the discriminatory network, i.e. to fool it into assessing a generated sample to be real, while the training objective of the discriminatory network is to minimize it's own error rate by becoming better at classifying generated samples as being fake. Both networks are usually trained via backpropagation and dropout algorithms, and samples from the generative model using only forward propagation. By utilizing this strategy of having two *adversarial* networks working against each other, it becomes possible to generate very realistic looking data samples, a task which is otherwise very hard to do[5].

3.2.2 CycleGAN-VC

StarGAN-VC is based on Cycle-Consistent Adversarial Networks [11], the mayor difference being that CycleGAN-VC only learns one-to-one mappings while StarGAN-VC does any-to-any mappings, a feature which is very important in the context of this project.

The most important aspects of CycleGAN-VC are:

- Take two different speakers, X (could be you, the reader) and Y (could be Barack Obama), and let $\mathbf{x} \in \mathbb{R}^{Q \times N}$ and $\mathbf{y} \in \mathbb{R}^{Q \times M}$ be their respective acoustic feature sequences of speech, where Q is the feature dimension and N and M are the lengths of the sequences. CycleGAN-VC aims to learn a mapping $G(\mathbf{x})$ that converts X (your voice) into Y (Barack Obama's voice). G is understood as the **generator** of a GAN [5].

- **Discriminator** D_Y, also referred to as real/fake discriminator, is introduced, the aim of D_Y being to predict whether acoustic speech data originates from $\mathbf{y}$ or is in fact generated by the mapping $G(\mathbf{x})$.

- **Adversarial Losses** for discriminator D_Y and generator G, as well as for inverse discriminator D_X and inverse generator F, are introduced, following the idea of a GAN:

$$\mathcal{L}_{\text{adv}}^{D_Y}(D_Y) = -\mathbb{E}_{\boldsymbol{y}\sim p_Y(\boldsymbol{y})}\left[\log D_Y(\mathbf{y})\right] - \mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{x}}(\mathbf{x})}\left[\log\left(1 - D_Y(G(\mathbf{x}))\right)\right] \tag{1}$$

$$\mathcal{L}_{\text{adv}}^{G}(G) = \mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{x}}(\mathbf{x})}\left[\log\left(1 - D_Y(G(\mathbf{x}))\right)\right] \tag{2}$$

$$\mathcal{L}_{\text{adv}}^{D_X}(D_X) = -\mathbb{E}_{\boldsymbol{x}\sim p_X(\boldsymbol{x})}\left[\log D_X(\mathbf{x})\right] - \mathbb{E}_{\mathbf{y}\sim p_{\boldsymbol{y}}(\mathbf{y})}\left[\log\left(1 - D_X(F(\mathbf{y}))\right)\right] \tag{3}$$

$$\mathcal{L}_{\text{adv}}^{F}(F) = \mathbb{E}_{\mathbf{y}\sim p_{\boldsymbol{y}}(\mathbf{y})}\left[\log\left(1 - D_X(F(\mathbf{y}))\right)\right] \tag{4}$$

The adversarial loss $\mathcal{L}_{\text{adv}}^{D_Y}(D_Y)$ represents a measure of how indistinguishable $G(\mathbf{x})$ is from real speech produced by X, and as the goal of discriminator D_Y is to correctly distinguish generated speech from real, D_Y seeks to minimize this loss. The same thing is true for inverse discriminator D_X and its adversarial loss $\mathcal{L}_{\text{adv}}^{D_X}(D_X)$.
As the goal of generator G is to produce speech indistinguishable from real speech, it seeks to maximize $\mathcal{L}_{\text{adv}}^{D_Y}(D_Y)$, while minimizing its own loss $\mathcal{L}_{\text{adv}}^{G}(G)$. Again, same thing applies to inverse generator F.

- **Cycle Consistency Loss** is introduced, which works to ensure that mapping from X to Y through $G(\mathbf{x})$ and then back from Y to X using the inverse mapping $F(\mathbf{y})$ results in close to the original data $\mathbf{x}$. This helps with preserving linguistic information of the source speaker, as there are infinitely many mappings that will produce the same output yet only few that inversely reproduce the input.

  $$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{\boldsymbol{x}\sim p_{\mathbf{X}}(\mathbf{x})}\left[\|F(G(\mathbf{x})) - \mathbf{x}\|_1\right] + \mathbb{E}_{\mathbf{y}\sim p_Y(\mathbf{y})}\left[\|G(F(\mathbf{y})) - \mathbf{y}\|_1\right] \tag{5}$$

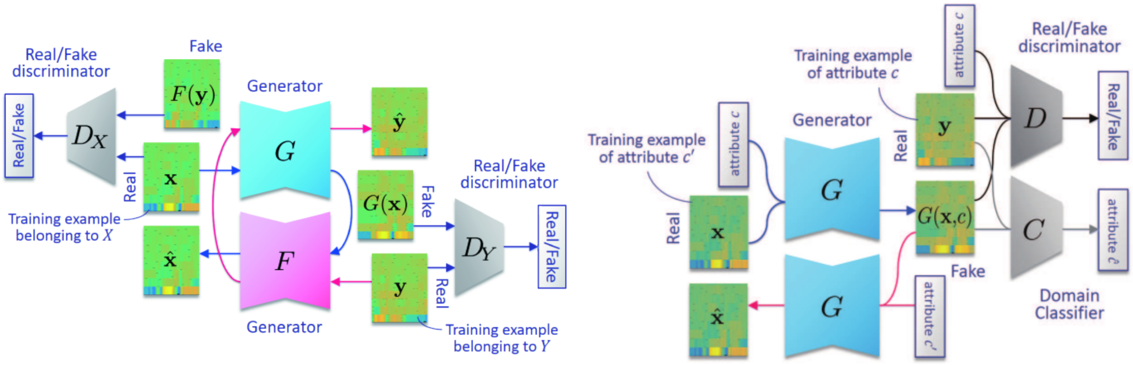
  CycleGAN overall seeks to minimize its cycle consistency loss $\mathcal{L}_{\text{cyc}}(G, F)$.

- **Identity Mapping Loss** is further introduced, working to ensure that the generator and thereby mapping G does not change its inputs ($\mathbf{x}$ in the case of $G(\mathbf{x})$), if the inputs given already resemble what they are being mapped to, i.e. $\mathbf{x}$ already resembles $\mathbf{y}$.

  $$\mathcal{L}_{\text{id}}(G, F) = \mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}\left[\|F(\mathbf{x}) - \mathbf{x}\|_1\right] + \mathbb{E}_{\mathbf{y}\sim p_Y(\mathbf{y})}\left[\|G(\mathbf{y}) - \mathbf{y}\|_1\right] \tag{6}$$

  CycleGAN seeks to minimize the identity mapping loss.

The Training structure of CycleGAN can be seen in figure 6a, in which generators G and F are updated repeatedly.



(a) CycleGAN training structure [13].
Generator G and inverse generator F are responsible for producing one-to-one mappings in CycleGAN-VC. Discriminators D_Y and D_X are used in training of the model by trying to identify speech as being real or generated by G and F.

(b) StarGAN training structure [13].
StarGAN only needs one generator, G, which is used for any-to-any conversions by assigning each mapping to a class c. Disciminator D seeks to identiify whether speech is real og generated by G. Domain classifier D seeks to identify which class/speaker real as well as generated speech belongs to.

Figure 6: CycleGAN and StarGAN training structures

3.2.3 StarGAN-VC

As this project investigates the possibility of mapping any voice into one without having to retrain the model itself, it is essential that the VC model is capable of handling multiple input speakers. StarGAN-VC [13] is able to perform any-to-any VC, and it does this on a basis of a modified CycleGAN and by adding another loss function. Instead of mapping strictly between X and Y, StarGAN-VC maps from a given acoustic feature sequence, say $\mathbf{x} \in \mathbb{R}^{Q\times N}$, to a given target speaker labeled c (class label), thereby generating the acoustic feature sequence $\hat{\mathbf{y}} = G(\mathbf{x}, c)$. StarGAN-VC keeps the idea of having a generator G and a discriminator D as with CycleGAN, but instead of having two of each, it has as many as there are speakers specified by c, represented as one-hot vectors. To take the different speakers (classes) into account, StarGAN-VC introduces a domain classifier C, which works to predict which speaker (class) the generated speech belongs to, and produces class probabilities $p_C(c|\mathbf{y})$ of $\mathbf{y}$.

Hereby, **Domain Classification Loss** is introduced:

$$\mathcal{L}^{C}_{\text{cls}}(C) = -\,\mathbb{E}_{c\sim p(c),\mathbf{y}\sim p(\mathbf{y}|c)}\left[\log p_C(c|\mathbf{y})\right] \tag{7}$$

$$\mathcal{L}^{G}_{\text{cls}}(G) = -\,\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),c\sim p(c)}\left[\log p_C(c|G(\mathbf{x}, c))\right] \tag{8}$$

The two domain classification loss functions (7) and (8) for classifier C and generator G respectively, take small values as classifier C correctly classifies which speaker (class) the actual speech sample $\mathbf{y} \sim p(\mathbf{y}|c)$ and the generated speech sample $G(\mathbf{x}, c)$ belongs to. Hence, StarGAN-VC aims to minimize the domain classification losses $\mathcal{L}^{C}_{\text{cls}}(C)$ with respect to C and $\mathcal{L}^{G}_{\text{cls}}(G)$ with respect to G.

Below are the updated loss functions for StarGAN:

Adversarial Losses for discriminator D and generator G:

$$\mathcal{L}_{\text{adv}}^{D}\,(D) = -\,\mathbb{E}_{c\sim p(c),\mathbf{y}\sim p(\mathbf{y}|c)}[\log D(\mathbf{y}, c)] - \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),\boldsymbol{c}\sim p(c)}[\log(1 - D(G(\mathbf{x}, c), c))] \tag{9}$$

$$\mathcal{L}_{\text{adv}}^{G}\,(G) = -\,\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),\boldsymbol{c}\sim p(c)}[\log D(G(\mathbf{x}, c), c)] \tag{10}$$

As in CycleGAN-VC, discriminator D seeks to minimize its adversarial loss $\mathcal{L}_{\text{adv}}^{D}\,(D)$ by correctly distinguishing real speech from fake generated speech. Generator G seeks to deceive D and thereby minimizing its adversarial loss $\mathcal{L}_{\text{adv}}^{G}\,(G)$.
Hence, D and G work to maximize each others adversarial losses.

Cycle Consistency Loss for generator G

$$\mathcal{L}_{\text{cyc}}\,(G) = \mathbb{E}_{c'\sim p(c),\mathbf{x}\sim p(\mathbf{x}|c'),c\sim p(c)}\left[\|G\left(G(\mathbf{x}, c), c'\right) - \mathbf{x}\|_{\rho}\right] \tag{11}$$

Again, minimizing cycle consistency loss is sought in order to preserve linguistic information, but as there no longer is a predetermined source and target speaker, generator G needs to hold both the direct and inverse mappings of a given set of speakers/classes c. This follows the principle of StarGAN that is the generator being able to perform any-to-any conversions, the training of which can be seen illustrated in figure 6b.
The target class from which the inverse mapping is performed is marked as c'.

Identity Mapping Loss for generator G:

$$\mathcal{L}_{\text{id}}(G) = \mathbb{E}_{e'\sim p(e),\mathbf{x}\sim p(\mathbf{x}|e')}\left[\|G\left(\mathbf{x}, c'\right) - \mathbf{x}\|_{\rho}\right] \tag{12}$$

Identity loss seeks to ensure that speech from source speaker c being put through generator G is unchanged if it already resembles the target speaker c'.

The full minimization objective of StarGAN-VC, in regards to G, D and C, in which λ_{cls} , λ_{cyc} and λ_{id} are regularization parameters:

$$\mathcal{I}_G(G) = \mathcal{L}_{\text{adv}}^{G}\,(G) + \lambda_{\text{cls}}\,\mathcal{L}_{\text{cls}}^{G}\,(G) + \lambda_{\text{cyc}}\,\mathcal{L}_{\text{cyc}}\,(G) + \lambda_{\text{id}}\,\mathcal{L}_{\text{id}}\,(G) \tag{13}$$

$$\mathcal{I}_D(D) = \mathcal{L}_{\text{adv}}^{D}\,(D) \tag{14}$$

$$\mathcal{I}_C(C) = \mathcal{L}_{\text{cls}}\,(C) \tag{15}$$

3.2.4 WORLD Vocoder

StarGAN uses 'WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications' [9] as a means of both extracting the prior mentioned feature representation of audio that StarGAN utilizes, as well as outputting the generated speech as actual sound.
The vocoder technology, on which WORLD is based, works specifically to synthesize human voice in such a way that the voice data can be inputted into and outputted from systems such as VC-models. WORLD works in real-time and produces good sounding audio, making it fitting for the task of voice conversion.

WORLD turns input data into StarGAN-useable acoustic feature vector sequences by computing the fundamental frequency F0 and a spectral envelope from the speech audio input described in more detail under section 2.4. The fundamental frequency F0 of the speech audio sample, defined as the inverse of the smallest period of a periodic audio signal [9], is estimated using the DIO algorithm [3]. The spectral envelope, computed on the fundamental frequency F0 and the waveform of the audio using the CheapTrick spectral envelope estimator [8], provides mel-ceptral coefficients which represent the speech audio as an acoustic feature vector.
After StarGAN conversion, the data is synthesized back into audible speech through the spectral envelope and WORLD re-synthesizing algorithm.

3.3 Variational Autoencoder

The second VC model that will be explored as a means of improving speech-to-text performance on dialect heavy speech originates from the paper “One-shot Voice Conversion by Separating Speaker and Content Representations with Instance Normalization” [23].

The proposed method of the paper will from here on be referred to as Variational Autoencoder or VAE for short. Besides being state-of-the-art and open source, the incentive for investigating VAE lies in its vastly different approach to the VC compared to StarGAN. VAE is in theory more easily implemented in the desired any-to-one VC setup, as the model specifically works for an any-to-any VC use case.

In this project the VAE is convolutional, as it uses convolutions to extract features from speaker audio represented as mel spectrograms.

The structure of a VAE can be seen in figure 7 below.

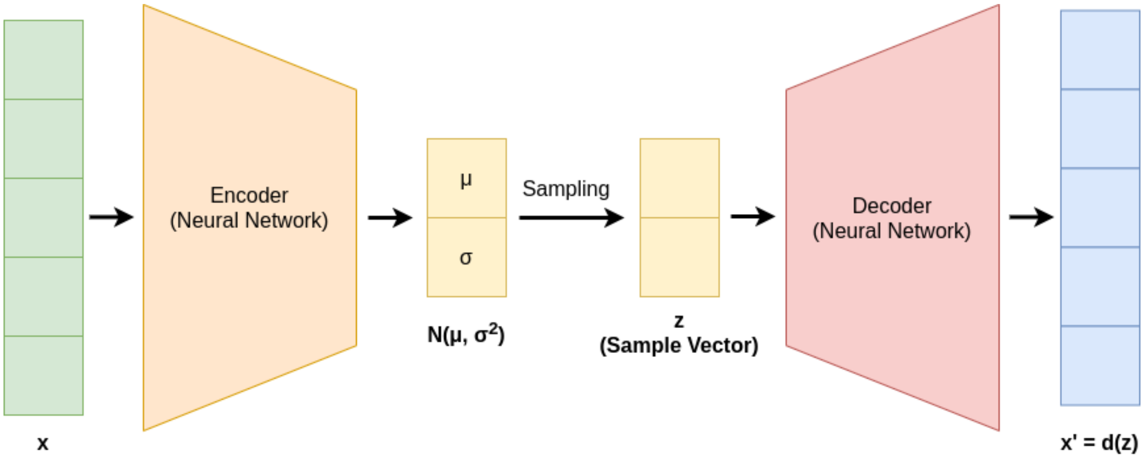


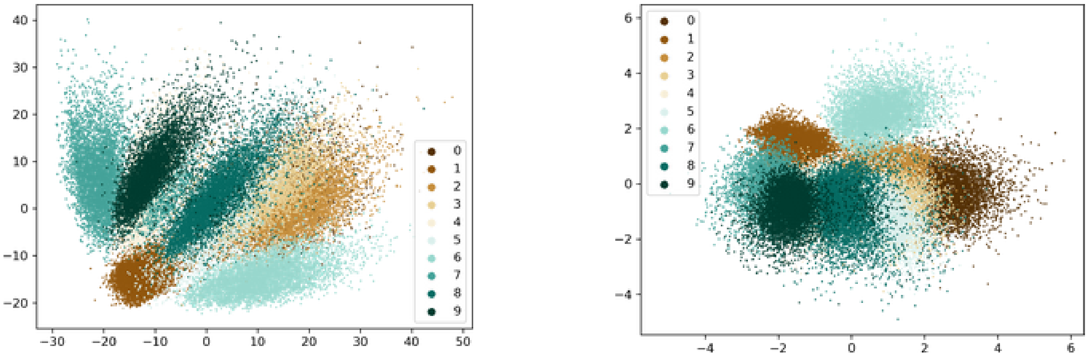
Figure 7: Variational Autoencoder Architecture: Input **x** is reduced by the encoder to a lower dimensional latent space **z** and then reconstructed by the decoder to the output **x’**

.

Both a standard autoencoder and a VAE work by taking an input, vector **x**, and reducing it to a lower dimensionality through encoding. This results in a latent representation of the input, which is passed to a decoder. The decoder then reconstructs an output, **x’**, from the given latent representation of the input. The latent space thereby works as a bottleneck which forces the autoencoder and VAE to learn an effective compressed representation of the input.

3.3.1 The Latent Space of VAE

It is the latent space which makes the VAE differ from a standard autoencoder. In a regular autoencoder, the latent space is represented as points or vectors which are often sparsely distributed and may not be continuous. This sparsity makes autoencoders well suited for compression problems and reconstruction problems. On the other hand, it creates challenges for generative purposes, because it becomes difficult to find latent values for which the decoder knows how to create an output [16]. The difference in the latent space can be seen below in figure 8.



(a) *Latent Distribution by Label for AE*

(b) *Latent Distribution by Label for VAE*

Figure 8: (a) Sparsely distributed latent space of a standard autoencoder. (b) Latent space of a VAE created from sampling from latent attribute distributions.
`https://thilospinner.com/towards-an-interpretable-latent-space/`

Because of this a VAE differs from a regular autoencoder by not having the encoder create the latent representation as points or vectors. Instead the encoder generates a probability distribution for each attribute in the latent space. However, using probability distributions for each attribute in the latent space is not sufficient for the latent space to be easy enough to work with. Therefore the VAE is optimized to force the latent attribute distributions towards standard normal distributions, by reducing the distance between the learned latent attribute distributions and the standard normal distribution. This is quantified in the objective function of the VAE by the Kullback–Leibler (KL) divergence [41], described later in this section. This helps the latent attribute distributions move as close to each other as possible while still being distinct, which can be seen in figure 8 (b).
The latent sample vector $\mathbf{z}$, is passed to the decoder by randomly sampling from the latent attribute distributions. As such, a VAE is not limited to recreation, but can generate new outputs by sampling from the latent space. This is useful for generative processes, such as converting a source speaker to another target speaker.

3.3.2 Objective Function

The objective function of a VAE, that is minimized during training, is a twofold quantity. The first half is equivalent to the objective function from an ordinary autoencoder, namely the reconstruction loss, but modeled as probabilities and expectations. As the name implies, it is a measure of how well the reconstruction created by the decoder compares to the original input data. In order to formulate this mathematically, the following definitions are introduced.

$\mathbf{X}$ is the collection of all acoustic segments in the training data and $\mathbf{x}$ is some acoustic feature segment.
E_s is the speaker encoder trained to generate the latent speaker representation $\mathbf{z}_s$, E_c the content

encoder, trained to generate the latent content representation $\mathbf{z}_c$ and D is the decoder. The idea is to infer the true distribution of $\mathbf{x}$ namely $p(\mathbf{x})$, by the conditional distribution $p\left(\mathbf{z}_c|\mathbf{x}\right)$ generated by E_c.
$p\left(\mathbf{z}_c|\mathbf{x}\right)$ is assumed to be a conditional and independent Gaussian distribution with unit variance i.e. $p\left(\mathbf{z}_c|\mathbf{x}\right) = \mathcal{N}\left(\mathrm{E_c}(\mathbf{x}), \mathbf{I}\right)$.
The reconstruction loss for a given speaker encoder $\theta_{\mathrm{E_s}}$, content encoder $\theta_{\mathrm{E_c}}$, and decoder θ_{D} then becomes:

$$L_{rec}\left(\theta_{\mathrm{E_s}}, \theta_{\mathrm{E_c}}, \theta_{\mathrm{D}}\right) = \mathrm{E}_{\mathbf{x}\sim p(\mathbf{x}), \mathbf{z}_c \sim p(\mathbf{z}_c|\mathbf{x})}\left[\left\|\mathrm{D}\left(\mathrm{E_s}(\mathbf{x}), \mathbf{z}_c\right) - \mathbf{x}\right\|_1^1\right] \tag{16}$$

This equation measures how much the decoded output differs from the given input, by taking the expected value over the l1-norm distance between the input vector $\mathbf{x}$ and the output vector.

The second term of the VAE objective function is the Kullback Leibler (KL) divergence loss between a diagonal multivariate Gaussian, and a standard Gaussian distribution[29]. As such, it is assumed that the latent attribute distributions $p\left(\mathbf{z}_c|\mathbf{x}\right)$ are normally distributed and independent, thereby giving the multivariate Gaussian it's diagonal property.
This is a measure of how well the encoded latent attribute distributions from the speaker input approximates a standard normal Gaussian. It can be quantified by:

$$L_{KL} = \frac{1}{2}\sum_{i=1}^{N} \sigma_i^2 + \mu_i^2 - \log\left(\sigma_i\right) - 1 \tag{17}$$

The KL divergence serves as a regularization term which encourages the modeled distribution to take on the shape of a standard Gaussian, because minimizing the L_{KL} is equivalent to moving σ_i towards 1 and μ_i towards 0.

The conclusive VAE objective function emerges as a weighted combination of the two terms.

$$\min_{\theta_{\mathrm{E_S}}, \theta_{\mathrm{E_C}}, \theta_{\mathrm{D}}} L\left(\theta_{\mathrm{E_S}}, \theta_{\mathrm{E_C}}, \theta_{\mathrm{D}}\right) = \lambda_{rec} L_{rec} + \lambda_{kl} L_{kl} \tag{18}$$

With weighted hyper parameters for the reconstruction loss λ_{rec}, and the KL divergence loss λ_{kl}.

The VAE is optimized by minimizing the reconstruction and KL divergence losses using gradient descent. Calculating the gradient for the KL divergence loss can be done without any additional steps, as the partial derivative can be calculated directly. This is not the case for the reconstruction loss, based on random samples, which doesn't have a gradient. To work around this, the *reparameterization trick* is used. It takes advantage of the fact that every Gaussian distribution can be expressed in terms of a standard Gaussian as:

$$\mathcal{N}\left(\mu, \sigma^2\right) \sim \mu + \sigma^2 \cdot \mathcal{N}(0, 1) \tag{19}$$

This make the generation of samples into:

$$z = \mu + \sigma^2 \cdot \epsilon, \quad \epsilon \leftarrow \mathcal{N}(0, 1) \tag{20}$$

Which factors out the randomness and makes it possible to calculate gradients for μ and σ, to use for optimization through backpropagation[16].

3.3.3 The VAE Approach

The leading idea in the VAE approach is that speech signals carry two kinds of information. Static information which is the speaker characteristics, that are considered time independent i.e. not changing over the span of a given utterance. And secondly, linguistic information, which is the meaning or content of an utterance and may change dramatically every time increment. In order to accommodate this entanglement the authors propose a variational autoencoder (VAE) model consisting of 3 modules as illustrated in figure 9.

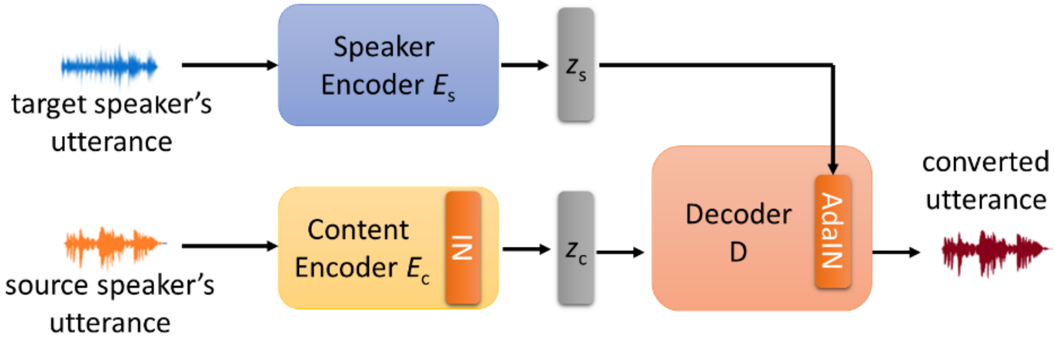


Figure 9: **VAE Model Overview:** E_s is the speaker encoder; E_c is the content encoder; z_s is the speaker representation; z_c is the content representation, and D is the decoder. IN is the instance normalization layer without affine transformation. AdaIN represents the adaptive instance normalization layer.
github.com/jjery2243542/adaptive_voice_conversion/blob/public/model.png

1. A speaker encoder that is trained to encode the speaker characteristics into a speaker representation.
2. A content encoder that is trained to only encode the linguistic information into a content representation.
3. Lastly a decoder which combines the two representations and synthesizes back the voice.

The content encoder archives its representation through instance normalization, which is the process of normalizing the data (multiple times) across the channels [1] as it passes through the convolutional network. This technique should in theory rinse the data of global information such as speaker characteristics. Furthermore, adaptive instance normalization is used in the decoder to shift and scale the content representation to match that of a given target [28]. The corresponding affine parameters utilized to do shifting and scaling is where the speaker encoder enters the picture, as it's tasked with learning and providing these parameters to the decoder.

[1] Channels can be thought of as feature representations uncovered by the convolutional layers

3.4 Implementation of StarGAN-VC

Implementing the two VC models, StarGAN described in section 3.2 and VAE in 3.3, is done using Python. The actual implementation of these VC models present an extra layer of complexity as the core theoretical principles of both, while being the backbone of the VC technology, require a set of coded practicalities in order to actually carry out voice conversion. All implementations are executed using python virtual environments containing all necessary modules in the correct version [19].

3.4.1 Core Components of StarGAN Implementation

The used implementation of StarGAN, as well as the one for VAE explained later under section 3.5, mainly consist of three main components:

- Preprocessing of speech data
- Training of the VC-model
- Converting speech data from source to target speaker

These components, which are implemented as modified versions of scripts provided through the original open source project github.com/liusongxiang/StarGAN-Voice-Conversion in the StarGAN-VC paper [13], and https://github.com/jjery2243542/adaptive_voice_conversion for the VAE paper[23] are described below.

Preprocessing of the input data is required for the model to be able to utilize it in training as well as conversion, and mainly takes part in the script *stargan_preprocess_spraakbanken.py*. The input data, described under section 2, are audio files in .wav format with different sampling rates, mainly 16 kHz which StarGAN is able to use directly, but in some cases 48 kHz recordings which is sampled down to 16 kHz before being preprocessed.
Using WORLD, described under section 3.2.4, implemented using the pyworld distributuon [39], the preprocessing first creates an overall summary 'speakerName_stats.npz' for each unique speaker based on the speech samples belonging to them. Every .wav are feature-summarized individually and further normalized by the mean and standard deviation obtaining in the overall summary of their corresponding speaker, saved as 'speakerName_audiofileXXX.npy'. These preprocessed files are now the acoustic feature sequence vector representations needed for the model to work with speaker data.

Training StarGAN is run via the main_spraakbanken.py script, though the actual training is performed in solver.py using PyTorch [37]. As explained earlier, in section 3.2, training the model involves repeatedly iterating over the Discriminator D and the Generator G.
Utilizing the preprocessed speaker data, training of StarGAN runs for 200,000 iterations, updating D and G throughout training with the overall goal of minimizing loss functions. The trained model parameters are saved separately for the discriminator D and generator G as .ckpt (checkpoint) files from which the VC-model is rebuild when taking on the conversion process.

Converting a seen or unseen source speaker requires preprocessing this speakers audio-data in the same regard as mentioned earlier. For StarGAN, the target speaker has to be designated from the original list of training speakers as the saved VC model parameters of G are defined as a one-hot encoded list of mappings. Each of the source speakers speech samples having been preprocessed into .npy files, will be fed into the generator G which converts and the outputs re-synthesized speech as .wav audio files with a 16 kHz sampling rate.
The script convertnew.py is a rather modified combination of preprocess.py and convert.py, which is in turn able to carry out the entire conversion process in one go.

3.4.2 Any-to-One StarGAN Considerations

For the sake of investigating this project's problem statement, any-to-one VC capabilities are crucial properties a VC model must posses. In the case of StarGAN, the training implementation used in this project is a replication of the original paper's training approach which is inherently an any-to-any training. However, this project only seeks to map to one designated target speaker, thereby making all other mappings redundant. Therefore this setup is inherently inefficient as its training is more computationally expensive having to train mappings that are redundant.
As just mentioned StarGAN-VC is not distinctly an any-to-one VC model, but is without further modifications able to perform these with surprisingly satisfying results. It cannot however, in the implemented state, convert to unseen target speakers as it is setup such that specifying target speakers from one-hot encoded trained speakers explained earlier is strictly required. General findings about StarGAN including any-to-one capabilities and attempts at improving these will be discussed shortly.

3.4.3 Pilot Training Experiments

In order to settle on a final composition for the optimal training data for retraining StarGAN, given the constraints of computational power, time and the available data, a series of pilot experiments were conducted. Among intuitions and ideas which needed to be investigated were:

- Performance scales well with the number of speakers. This originates from the notion that, the more diverse a training set is, the broader and more of a general understanding StarGAN receives, thus accommodating any-to-one conversions.

- Performance scales well with data per speaker.

- Any-to-one performance can be accommodated or improved by having an excessive amount of data on the target speaker, thus learning a strong mapping to that one speaker.

To investigate the above mentioned the following training pilot experiments were conducted and evaluated on, both during the training, seen and unseen data:

- On **few** speakers with **few** samples. 10 speakers with 312 samples per speaker.

- On **several** speakers with **few** samples. 25 speakers with 312 samples per speaker.

- On **few** speakers with **many** samples. 10 speakers with 987 samples per speaker.

- On few speakers with many samples WITH one additional speaker with an **abundance** of samples. 10+1 speaker with respectively 987 and 3800 samples per speaker.

- On **very few male only** speakers. on 5 male speakers with 987 samples per speaker.

It was found that the basic setup with few speakers and few samples performed very well on both seen and unseen data. This configuration will be the baseline by which the other pilot experiments will be compared and evaluated. Adding more speakers only improved performance slightly. Adding more data per speaker had a considerably greater positive impact on performance than adding more speakers. Performance dropped when the target speaker had an abundance of training data. Finally the training only consistent on very few male speakers performed worse than the baseline but still surprisingly well even when converting female speakers.
Regarding the the choice of the final training set for StarGAN, it was of highest priority to include as much data per speaker as possible, thus making the obvious choice to go with speakers from the Spraakbanken-test data which counted 987 samples per speaker. From thereon as many speakers as possible were included, capping out due to computational limitations, at 25. The 25 speakers were picked as diverse as possible from the 56 possible speakers in the Spraakbanken-test set.

3.4.4 Final Training Configurations of StarGAN

The setup used for training the StarGAN model used in investigating the research questions has the configuration described below. All settings, aside from the utilized data, are default settings from original implementation.

25 training speakers, 13 male and 12 female, from Spraakbanken-Test data set containing 987 recordings per speaker. The distribution of these were;
Fyn: 4,
Østjylland: 4,
Nordjylland: 2,
Vestjylland: 3,
Vest- og SydSjælland: 5,
Sønderjylland: 2,
København: 5
Target speaker is r6110050, a 43 year old man born and raised in København.
The number of model iteration used in research questions is 200,000, which is also the maximum iteration run in training.
The batch size is 32, designating the amount of training data samples (recordings) are collated together into batches in Pytorch when training the model [40].
Learning rates of D and G are 0.001.
Learning rate decay kicks in after 100,000 iterations, meaning halfway through training the learning rate drops a small amount every 1000 iterations.
Weights of domain classification loss and cycle consistency loss are set to 10.

3.5 Implementation of Variational Autoencoder

The components are modified versions of original scripts provided through the project `https://github.com/jjery2243542/adaptive_voice_conversion` [17]. Only slight changes have been made to these scripts. These include renaming some functions, removing unused code, and adding some comments, however the overall logic and settings in the project code has been preserved.

Preprocessing of the Danish speaker utterances is the first step to prepare the data for training the VAE model. Each speaker has a set of wav-files, which are created with 16kHz frequencies. These .wav files are processed by extracting mel spectrograms of each file using the python library Librosa. The spectrograms are held in memory as values in a hash map where the corresponding wav file names are the keys. This hash map consumes a lot of memory, due to the size of the spectrograms, and thereby limits the amount of speakers that can be processed. All spectrograms are normalized by subtracting the mean and dividing by the standard deviation
In the second step, a segment size is defined to be 128. It determines how large segments to sample from the generated spectrograms. All training speaker utterances are filtered, such that only .wav files with more than the defined 128 spectrogram segments are used.
From the remaining .wav utterance files, a total of 100,000,000 sample segments are selected at random to be used for training.

Training is run for 200,000 iterations saving a model each 500 iteration. The lowest error model is then chosen to be used for conversions.
The reconstruction loss is calculated using the L1Loss function from pytorch, which measures the mean absolute error between two given input. The KL loss is calculated using pytorch, by implementing equation 17 presented in section 3.3.2.

Conversions are made using one male target speaker with id r6110050, and one female target speaker with r6110032 chosen from the training set. Any number of unknown source speakers, that have not been part of the training set, can be chosen and the VAE model then converts the source speakers to the designated target speaker in a 16kHz .wav audio format.

3.5.1 Encoder and Decoder Architectures

Below are some detailed illustrations regarding the architectures of the encoders and decoder used by the implemented VAE. For each architecture a description is made about the main aspect in regards to how it supports the voice conversion strategy.

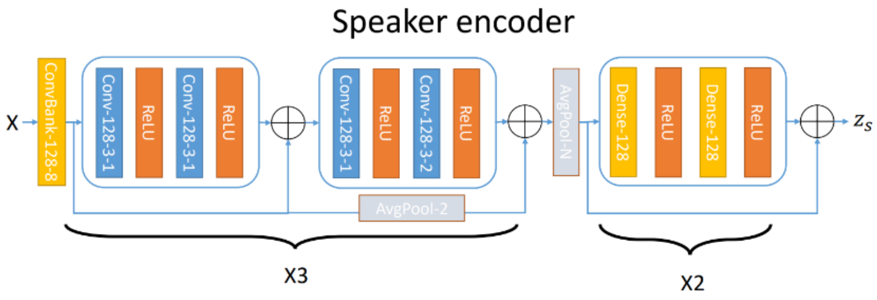


Figure 10: **Architecture of the speaker encoder** used by VAE to extract the speaker representation. [17]

The use of average pooling in the speaker encoder, figure 10, is worth noticing. It is applied as to enforce the speaker encoder only to learn global information from the speakers and not overfit to the linguistic content of the utterances [17].

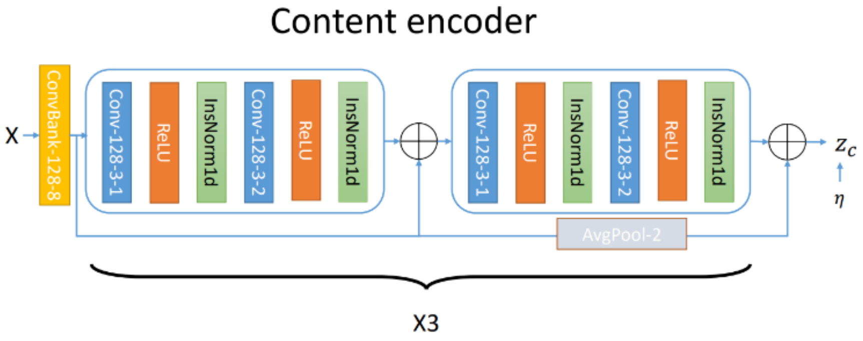


Figure 11: **Architecture of the content encoder** used by VAE to extract the linguistic content representation. [17]

For the content encoder, figure 11, instance normalization layers are important. As explained earlier it helps to rinse the audio input for speaker characteristics, thereby leaving the linguistic content.

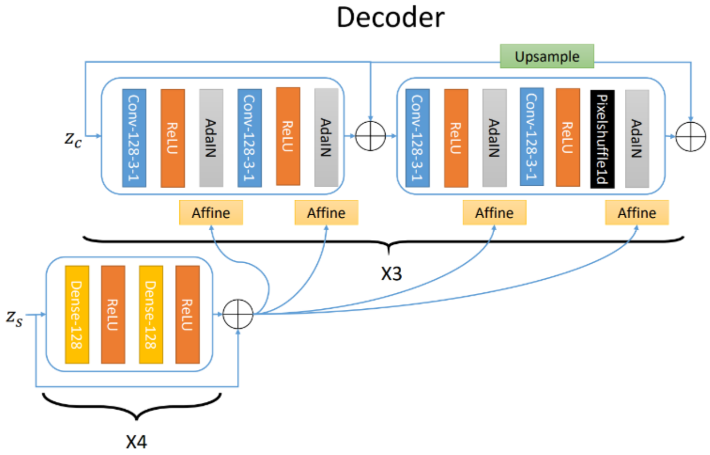


Figure 12: **Architecture of the decoder** used by VAE to translate the encoded speaker and content representations into a given target speaker. [17]

Lastly in the decoder, figure 12, the AdaIN layers are important, as they receive the speaker representations through the affine layers. This makes it possible for the AdaIN layers to transfer the speaker styles onto the linguistic content in the decoder.

3.5.2 Pilot Training Experiments

To investigate the capabilities of VAE and decide on a choice of settings, a series of pilot experiments were conducted. The investigation was limited by time constraints and by technical issues with running the original VAE code. However, the following tests were made:

- Performance on **few** speakers with **many** samples. 10 speakers with 987 utterances each.
- Performance on **many** speakers with **fewer** samples each. 614 speakers with 312 samples each.
- On **several** speakers with **few** samples. 25 speakers with 312 samples each.

For all experiments a close to equal distribution of speaker dialects and speaker sex was ensured. Neither the difference in the number of speakers nor the number of utterances per speaker had a noticeable impact on the quality of the voice conversion. Therefore a decision was made to implement settings for the 25 speaker set up, as it provided a compromise between the low 10 speaker and large 614 speaker amounts.

3.5.3 Final Training Configurations of VAE

The setup used for training the VAE model used in investigating the research questions has the configurations described below:

The utilized data, are similar to the ones used in the StarGAN pilot experiments:
25 training speakers, 13 male and 12 female, from Spraakbanken-Test data set containing 987 recordings per speaker. With dialect distributions:
Fyn: 4,
Østjylland: 4,
Nordjylland: 2,
Vestjylland: 3,
Vest- og SydSjælland: 5
Sønderjylland: 2,
København: 5
The used training batch size is 256.
The applied ADAM optimized has following settings:
Learning rate of 0.0005, β_1 as 0.9, β_2 as 0.999, and a weight decay of 0.0001.
Finally the weights for the objective function are λ_{rec} set to 10 for the reconstruction loss, and λ_{KL} set to 0.01 for the KL Leibler loss.

3.6 Speech to Text Model

As the name suggests, the speech to text technology is essentially a system that translate audio data of speech into its written representation. There are many different frameworks and approaches to performing speech to text for many languages, yet most are centered around English and thereby only few are able to translate danish speech into text.

3.6.1 danspeech

The term danspeech covers a number of STT modelsdanspeech developed by Martin Carsten Nielsen and Rasmus Arpe Fogh Jensen as part of their Master's thesis with the motivation of providing Danish developers with an open-source alternative to the few STT models capable of handling Danish speech. danspeech is distributed as a python package and mainly based on Pytorch [38]. danSpeech's Documentation [26] can be found on their github page.
This project seeks to utilize VC as a mean to improve danish danspeech's STT performance. Below is a brief description of the features and techniques used by danspeech, as well as this papers implementation and configuration of danspeech models.

Framework: The framework on which danspeech's models are build are end-to-end Deep Speech 2 models [6]. Deep Speech 2 is a deep learning framework specifically designed for speech-model training in different languages with a focus on high computational capacity for quicker retraining of models.
In the case of Deep Speech 2, and thereby danspeech, performing STT involves a number of steps build up around the concept of a recurrent neural networks (RNN) and its different sets of hidden layers:
First step is to extract relevant speech feature sequences from the audio signal using convolutional layers.
Next step is done using recurrent layers which work to propagate information through such feature sequence. This propagation outputs a matrix representation of each sequence, that can be used to train the RNN and be decoded in order to output the text that was sought after in the first place. A fully connected layer is finally used for actual classification of words.

The final step thereby becomes outputting the STT-translated speech, which is done using a decoder, in this case Connectionist Temporal Classification (CTC) decoder. With such procedure performed by an STT model, speech can then be translated to text [6].
The layer structure used for the STT model can be seen in figure 13.

CTC: The danspeech models are, as with Deep Speech 2, trained with a CTC loss and uses CTC as decoder for outputting text. **CTC loss function** calculates a loss for each training sample, in which the RNN seeks to maximize the probability of correct speech classification, i.e. minimize CTC loss, for optimal performance. After training the RNN using CTC loss, **CTC decoding** is used to calculate the most likely text given the speech audio input [21].

Pre-trained danspeech models: Already trained danspeech STT models are available for direct plug-and-play use, these being trained mainly on the Danish dataset Spraakbanken by NST [36] also used by this project, and further on danspeech's own 'DanSpeech' data set containing around 1000 recordings [26].

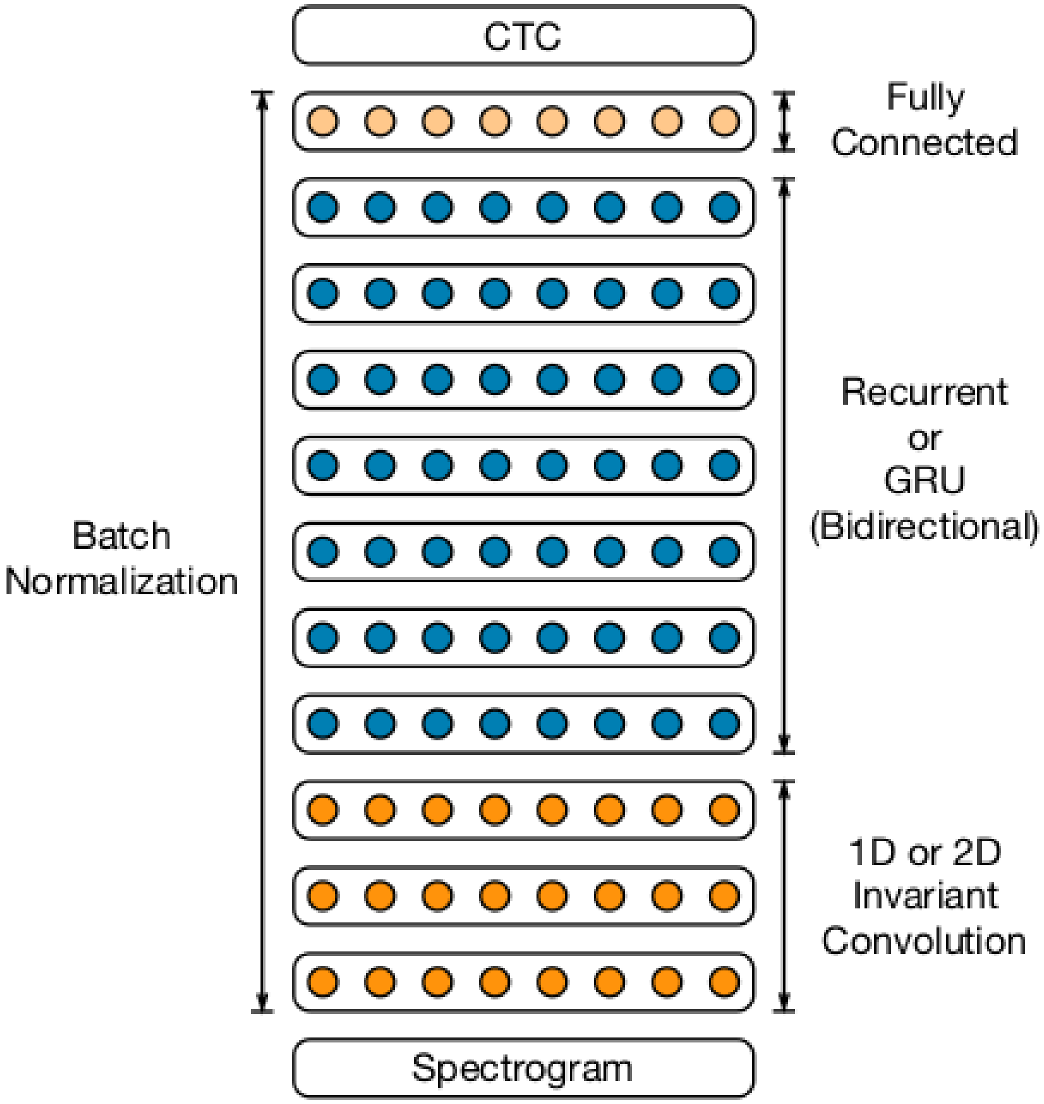

CTC
Fully Connected
Recurrent or GRU (Bidirectional)
Batch Normalization
1D or 2D Invariant Convolution
Spectrogram

Figure 13: Deep Speech 2 RNN hidden layer setup consisting of three convolutional layers, seven recurrent layers and finally one fully connected layer before the CTC decoder.

With exception of the third research questions **all** applications of a danspeech SST model, in this paper, will be done by their best pre-trained model, namely the *DanSpeechPrimary* model.

Parameter Tuning of danspeech Models

In order to employ a danspeech model to its full capabilities the method *update_decoder* has to be called on the model. This method takes 3 hyper parameters, *alpha*, *beta* and *beam_width*. A random search was conducted with the *DanSpeechPrimary* model in order to optimize these parameters. The random search was carried out on a data set consisting of 2 male speakers and 2 female speakers, with 5-6 voice clip per speaker, each voice clip containing one sentence. All sentences were unique and contained on average 22 words, resulting in a combined 464 words across all speakers. The data originated from audio books and was of excellent quality, with no noise and completely clear, well separated pronunciations of the individual words, spoken in almost dialect deprived voice.

The parameters were optimized with respect to achieving the best possible word error rate (word error rate is introduced in its entirety in 3.7.3). The best word error rate found by random search was 0.263 with corresponding parameters $alpha = 0.65$, $beta = 0.65$ and $beam\_width = 227$. These parameters will be used through out the paper in all applications of a danspeech model.

Retraining

Models from danspeech can be retrained, such as proposed by the third research question, and is an integrated part of danspeech. This means that retraining danspeech to fit data that is a voice converted version of original training data is possible. For the purpose of investigating the third research question, retraining of danspeech will also have to be done on non-voice converted data as this will serve as comparison for the voice converted danspeech variant.

Both retrainings will be performed using the entire Spraakbanken-train set consisting of 614 speakers with 312 utterances per speaker.

3.7 VC Evaluation Methods

This paper will not be evaluating the produced VCs by conventional VC evaluation methods but will instead draw on methods from the STT literature. The reason behind the choice of evaluation method and what exactly it involves will be elaborated in the following sections.

3.7.1 Popular VC Evaluation Methods

Within the VC literature the most common performance metric the authors of a proposed model concern themselves with is the Mean Opinion Score, MOS [33]. MOS is a 5-point scale with 5 being excellent and 1 being bad and was originally used to evaluate the quality of telecommunications. MOS is by nature subjective as the scoring is performed by human test subjects. In practice there are a very strict set of rules regarding everything from room dimension to who may participate as test subject in order to make the MOS as objective as possible and thus comparable. Letting this performance metric rely exclusively on human perception does make sense in most VC applications, as the goal of the VC in the first place often is to satisfy a certain human standard or even completely deceive the human ear.

Among other popular VC evaluation methods are the AB-test and similarity test. The AB-test is used to investigate the relative performance between two models, while the similarity test is used to asses how convincing the VC of a model is. Like the MOS these tests as well, need to be conducted using human test subjects. Unlike the the MOS both of these test cannot be used for determining how the a VC model places in the global landscape of the competition.

3.7.2 A Different Approach to VC Evaluation

The above described VC evaluation methods are primarily centered around a subjective humans perception of produced voice conversions.

Due to this study's approach to focusing on using VC as a means of improving speech-to-text (STT) systems, the evaluation of the VC will be done indirectly by considering the performance of the resulting STT which can be asserted objectively without involving human opinions and subjectivity. Furthermore, the success criterion for a good VC model in the so far stated methods all comes down to how good/real the produced conversions are perceived by human listeners. In the grand scheme of this project, the human perception of conversions is thought to be only partly relevant with respect to investigating the second research question and is hypothesized to be entirely irrelevant in the third and final research question. To elaborate this notion further:

With regards to the second research question, the humanly perceived quality of conversions will likely affect how well danspeech is be able to interpret and transcribe the voice converted speech sample. The standard danspeech STT model is trained on and optimized for human voices, i.e. the more real and human a speech sample sounds the more likely danspeech is to transcribe it correctly. This is however not the case for the third research question. When retraining danspeech exclusively on voice converted speech with a single target speaker, the 'humanness' of the speech essentially becomes irrelevant. After retraining, danspeech will only be trained on one voice, and in theory it will not care whether unseen test samples sound human as long as they are consistent with the target voice it has been trained on.

In both research questions, the VC only serves as an intermediate; a tool with which the STT of danspeech can be improved. The success of the VCs will indirectly be asserted with respect to the final output of the danspeech STT.

3.7.3 Word Error Rate - WER

Where the VC literature has an abundance of performance measures, the STT literature primarily concerns itself with a single quantity, the word error rate, WER for short [44]. WER is a similarity measure between two strings of words, what was actually said (the true transcript) and the transcript produced by the STT system (the proposed transcription). WER is given by:

$$WER = \frac{S + I + D}{N} \tag{21}$$

- S is the number of substitutions, i.e. where correct words have been replaced with an incorrect word.
- I is the number of insertions, in which words not present in the true transcription appears in the STT-proposed transcription.
- D is the number of deletions where some correct word is completely absent in the proposed transcription.
- N is the total number of words in the true transcription.

The output for a given WER calculation will generally be on a scale from 0 to 1, and can roughly be interpreted as a percentage. A WER of 0 is the best possible score and denotes that the true transcript and the proposed transcript are perfectly identical, while a WER of 1 indicates that the proposed transcript is completely wrong.
With this test, the different VC model apporaches can be evaluated objectively against each other. The WER test will be carried out using the python library jiwer [30].

Although WER is the industry standard, it bears some shortcomings:
Firstly, due to the formulation of the WER equation, the WER can exceed 1. This should be disregarded and any instances where the WER assumes anything greater than 1 it should be considered 1,i.e. completely incorrect.
Secondly, the WER score does not take into account the complexity of a transcription, only the amount of substitutions, insertions and deletions. Because of this, two differently converted versions of the same speech sample with the same amount of these failures can be close in WER score even though the complexity of the correctly transcribed words was different.
Thirdly, based on findings from pilot experiments with WER, it became apparent that longer strings receives worse WER than if they are split into smaller bits weighted by their length. It was not investigated whether this property was specific to this papers implementation of WER, namely the jiwer python library.

3.7.4 Mcnemar Test

In order to determine the significance of difference in WER-score performance, the McNemar test will be employed [32]. The McNemar test is popularly used within the classification literature to determine statistical significance between the performance of two classifiers in a paired test scenario, i.e. where they are given the same test data.
The use of a McNemar test makes intuitively more sense when the STT system is considered as a classifier which takes some input speech sample and produces some label in the form of an output

transcription. In this study, each individual word of the output will be considered one classification task, which can either be correct or incorrect. This also implies that when conduction the McNemar test, only a few sentences are needed to produce significant results, as a few sentences easily contain upwards of 100 words corresponding to 100 individual classification tasks. The test statistic of the McNemar test is given by the following expression:

$$\chi^2 = \frac{(b-c)^2}{b+c} \tag{22}$$

Here, c is the number of instances where classifier 1 is correct and classifier 2 is not, while b is the number of instance where 2 is correct and 1 is not. The intuition behind this choice of parameters is that instances where both models are correct must have been trivial and instances where both are wrong must have been unreasonably difficult, thus leaving only b and c to be of interest.

χ^2 is assumed to follow a chi-sqaured distribution with one degree of freedom if the number for discordants (the instances of b and c) is sufficiently high. This criteria should be met due to the choice of test data.

Further assumptions about the the McNemar test are: Firstly, the two variables being compared must be dichotomous, meaning they can take on one of two possible outcomes. In this setting, the variables are the classifiers and they can either be correct or incorrect.

Secondly the variables must be mutually exclusive which is insured by the formulation of b and c.

Lastly the test data must be randomly sampled from the the population of interest. This assumption is the most difficult to satisfy, as hand picking the data introduces a subconscious bias which might have influence on the choice of test data.

3.7.5 Sanity Check

A sanity check is a term used to describe a test that merely seeks to verify whether claimed functionality of a system can in fact be true.

In the case of this project, a sanity check will be made when investigating the first research question, as the purpose of it is to verify that the two VC models, StarGAN and VAE, in fact work as claimed and are capable of many-to-one, zero-shot conversions. These sanity checks will be based on the authors interpretations of VC-model-conversions made with the purpose in mind of showcasing different possible strengths and weaknesses the models might have. A list of the 25 speakers used to train VAE and StarGAN for the sanity check can be found in appendix A.

4 Results

4.1 First Research Question

How well can state-of-the-art voice conversion results from StarGAN-VC [13] and Instance-normalization [17] VC models be reproduced for many-to-one, zero-shot voice conversion scenarios?

4.1.1 Sanity Check of Voice Conversion Models

Investigating this project's overall problem statement and following research questions relies heavily on being able to perform satisfying voice conversions. This does not strictly mean that voice conversions have to sound good to the human listener, as this project specifically investigates it's effect when implemented with an STT model. However, it does mean that the problem statement and idea behind the project emerges from being able to perform good voice conversions based on available state-of-the-art VC models, and in this regard it is of interest to be able to reproduce such state-of-the-art results, specifically under the any-to-one zero-shot condition that this project requires.

With the implemented versions of StarGAN and VAE, conversions made utilizing these models were evaluated in a sanity check, described under 3.7.5. The check sought to verify whether the performances of the implemented VC models actually matched the ones claimed by their authors, and to investigate how speaker dialects were affected by the voice conversions.
The sanity check was based on conversions from a number of different Danish source speakers as well as a couple of English speakers. All source speakers were converted both to a male target speaker as well as a female target speaker. These conversions were evaluated subjectively focusing mainly on the linguistic preservation and the ability to convert speaker style. The utilized test conversions for StarGAN and VAE can be found in the gitlab project folder: /results/sanity_check.

4.1.2 StarGAN

The implemented version of StarGAN showed promising results, both for seen and unseen source speakers, as well as for inter and intra gender conversions.
Test conversions, which can be heard via above mentioned link, showcased StarGAN convincingly generating the target speakers voice, both male and female, and maintaining a sound quality that was for the most part not interfering with the listeners perception of the linguistic content. Conversions made between speakers included in training, i.e. non-zero-shot conversions, seemed only slightly more clear than zero-shot conversions which performed very well.
The converted speech samples, now being completely different in speaker style, matching target speaker style instead of source, were not changed in terms of actual dialectic characteristics. Because of this, the sanity check found that while the conversion process changed acoustic features of a speaker heavily, the actual dialect-defining characteristics remained and so StarGAN cannot be said to convert dialect but only speaker style.
An apparent negative trend for zero-shot conversions was a tendency to, within the same speech sample, have volume fluctuations. It was noticed that this trend was most pronounced on conversions made on data gathered separately from Spraakbanken.
StarGAN was further able to produce inter-lingual conversions, meaning an English source speaker could be converted to a Danish target speaker's voice, and though the quality of conversion in such

case was noticeably lower than that of intra-lingual, it did in fact work.

Figure 14 shows losses during training of the StarGAN model.

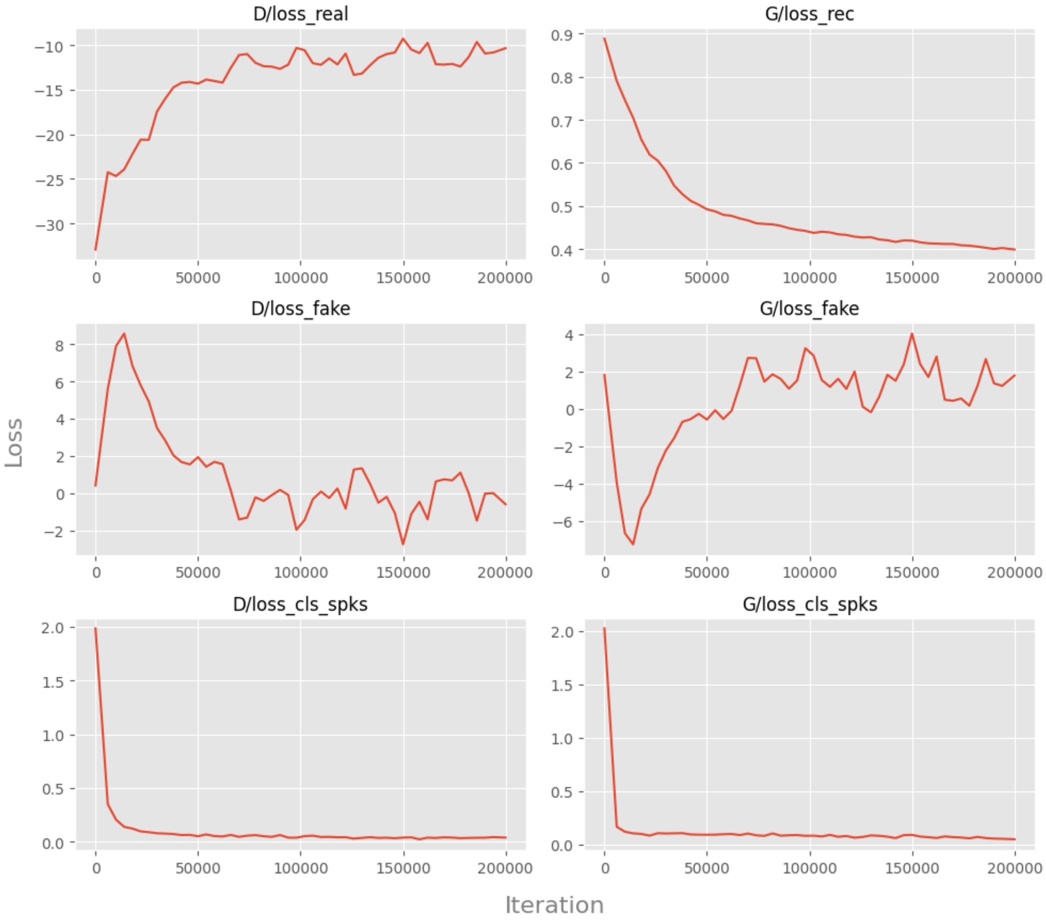


Figure 14: Discriminator and Generator function losses recorded during training of StarGAN on 25 Spraakbanken speakers which was used in the sanity check.
D/loss_real is the adversarial loss of *D* measured in cases where speech was real. D/loss_fake and G/loss_fake are adversarial losses for cases where speech was generated. G/loss_rec is cycle consistency loss for G. D/loss_cls_spks and G/loss_cls_spks are classification losses for D and G respectively.

4.1.3 VAE

The implemented version of VAE was sanity checked on the same source speaker data as StarGAN. The results of the VAE voice conversions were deemed of poor quality.

Part of the linguistic content was not decoded properly, resulting in sounds with no actual linguistic meaning for some utterances. The generated speaker style was consistent across the conversions, but the style did not match the target speaker for any of the conversions. The converted speaker style ended up being too noisy, dark voiced and mumbling, making it sound far from a real human voice.
The conclusion of this sanity check was that the implemented VAE model was not well suited for the voice conversions.

Figure 15 shows the losses during training of VAE.

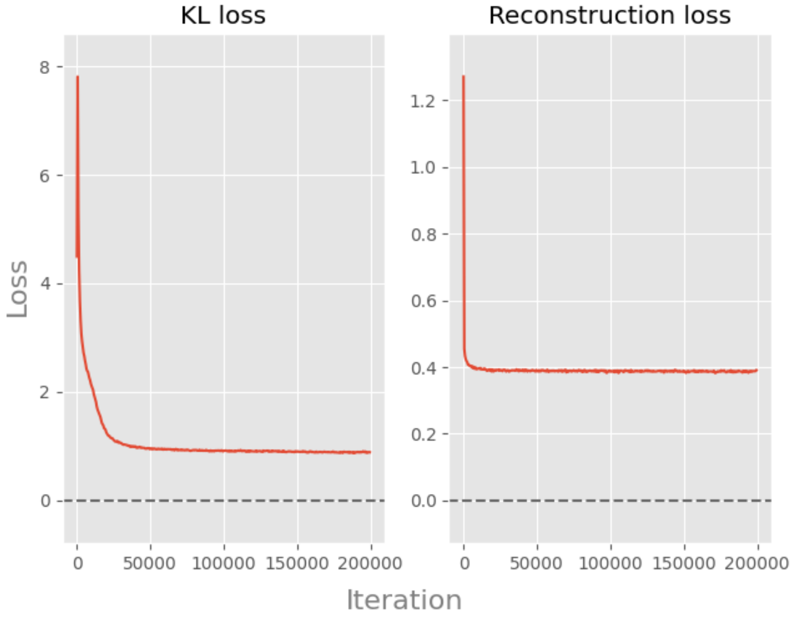


Figure 15: Recording of loss output from the training of VAE on speakers from /results/sanity_check/original.

Within fifty thousand iterations the KL loss and reconstruction loss, described under 3.3.2, quickly reached their optimum and stayed there. This behavior is worrying in the sense that the VAE in the original paper was trained on 200,000 iterations, supposedly because the authors continued to see improvement over this many iterations. The fact that this papers implementation stagnates within so few iterations seems to strongly suggest that something is off. This behavior could be a result of overfitting in the VAE model.

4.2 Second Research Question

How does the danspeech speech to text transcriptions perform when applying voice conversion models compared to using no voice conversion?

4.2.1 danspeech Pre-Trained

The overall problem statement proposes the idea of improving STT by having the STT model exclusively trained on a single common voice. The setup where new data is converted to a common voice, should help the STT model, which then only has to transcribe the one common voice it knows.

In order to do preliminary investigations of the problem statement, it was of interest to evaluate the impact of voice converting an input prior to running it through a standard danspeech STT transcription.
The danspeech technology is described more in detail under section 3.6. Its pre-trained STT models are supposedly better at transcribing Danish voices from the Storkøbenhavn as these are represented more strongly in the available training data. In this respect, the second research question of this project sought to convert voices into a common danish voice from that area. And so, the male speaker with id r6110050 from Spraakbanken was chosen as the target speaker. Then it could be checked if the pre-trained danspeech models already would benefit from only having to transcribe speech that was converted. For the experiment the Spraakbanken *test* data was used.

4.2.2 Results of Second Research Question

Using Word-Error-Rate (WER), described under section 3.7.3, as a measure of transcriptive performance, the standard pre-trained danspeech STT model was evaluated on the same speaker utterances in original voices, StarGAN-converted voices and VAE-converted voices. By application of the McNemar test, described under section 3.7.4, all WER results within the same dialect or sex were found to be significantly different from one another with regards to a 0.05 significance level. The particular word error rates are presented in table 1.

The test showed that the pre-trained danspeech model performed best on original speaker voices, consistently followed by StarGAN-converted voices and then VAE-converted voices. These WERs can be seen in figure 16 displaying WER for the different dialects and figure 17 for different sexes. The easiest dialects to transcribe was from Syd- og Vestsjælland and Storkøbenhavn. The most difficult dialect to transcribe was from Sønderjylland with a small gap between performances of the three speech data types.

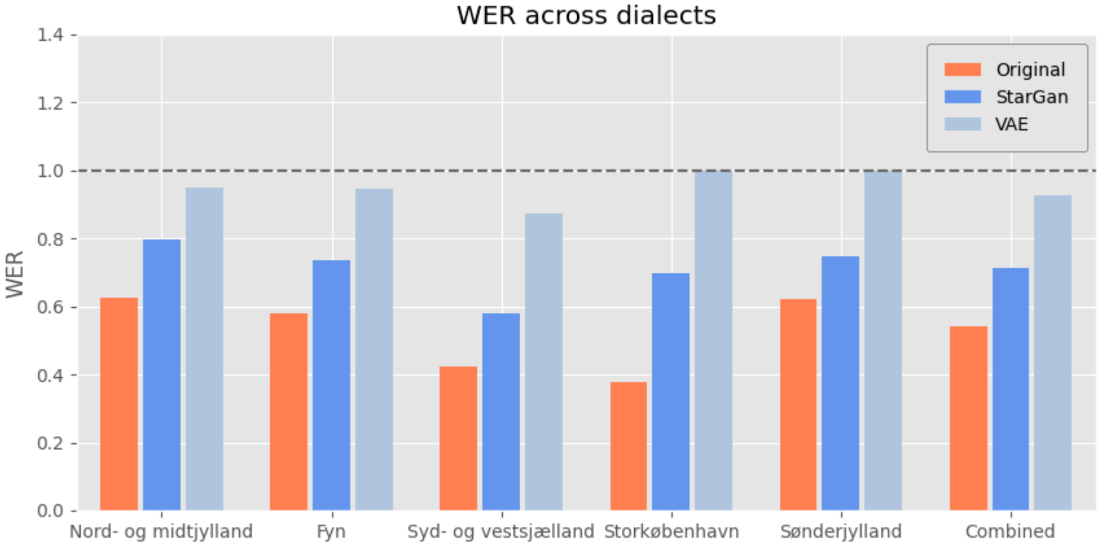


Figure 16: WER of danspeech for transcribing original, StarGAN converted and VAE converted speech samples categorized for every included dialect.

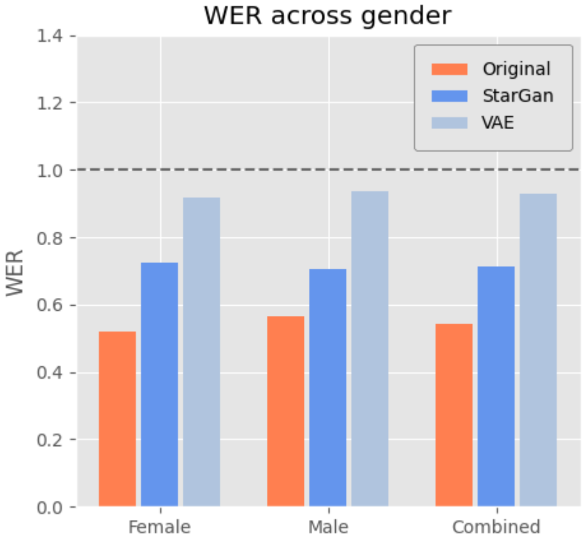


Figure 17: WER of danspeech for transcribing original, StarGAN converted and VAE converted speech samples categorized for speaker sexes.

The most noticeable trend in the test results was a moderate consistency in relative performance difference between the WER of danspeech on StarGAN converted speech and original, across dialects and sex. Except for Storkøbenhavn, the transcriptions made on original data were consistently around 0.15-0.25 better in WER score

	Nord- og midtjylland	**Fyn**	**Syd- og vestsjælland**	**Storkøbenhavn**	**Sønderjylland**	**Female**	**Male**	**Combined**
Original	0.578	0.626	0.422	0.378	0.62	0.521	0.566	0.542
StarGAN	0.735	0.797	0.578	0.698	0.747	0.723	0.706	0.715
VAE	0.944	0.951	0.872	1.0	1.0	0.919	0.938	0.928

Table 1: Word Error Rates for Spraakbanken *Test* Data: The word error rates are measured across five dialects, female and male sex, and for the full combined speaker data. The Speech data used is the original speaker utterances, utterances converted by StarGAN and conversions made by VAE.

Transciptions by danspeech made on VAE converted data scored an overall WER of 0.928. With the worst WER practically achievable being 1, the test further emphasizes the claim of VAE having very poor VC capabilities (as described in the first research question under section 4.1), leading directly to danspeech being virtually unable to transcribe speech correctly. The significant difference in performance between the two VC models proved an evident superiority of StarGAN in relation to pursuing the problem statement. In the light of this finding, it was determined that VAE would not be utilized in retraining the danspeech STT model when investigating the third research question.

4.3 Third Research Question

How does the danspeech speech to text translation perform when voice converted input is provided to a pretrained danspeech model compared to a danspeech model retrained on voice converted data?

4.3.1 Retrained danspeech Model

As the results from the second research question suggested, it is not possible to achieve acceptable STT translations with voice converted data using pre-trained danspeech models.
As such, it was important to conduct experiments on retrained versions of the danspeech models. These experiments were performed using the full Spraakbanken training data set, 614 speakers with 312 utterances each. Two danspeech models were evaluated: One retrained on the original Spraakbanken data using the conventional danspeech setup. The second model was retrained on voice converted Spraakbanken data using StarGAN. The StarGAN STT setup was used for testing, where the input speaker audio was converted before sending them to danspeech and calculating the WER. The resulting WER scores of each setup are compared across dialects in figure 18 and across gender in figure 19 below.
As previously explained VAE was not included in the experiments, as the final quality of its voice conversion performed too poorly in the WER tests of the second research question.

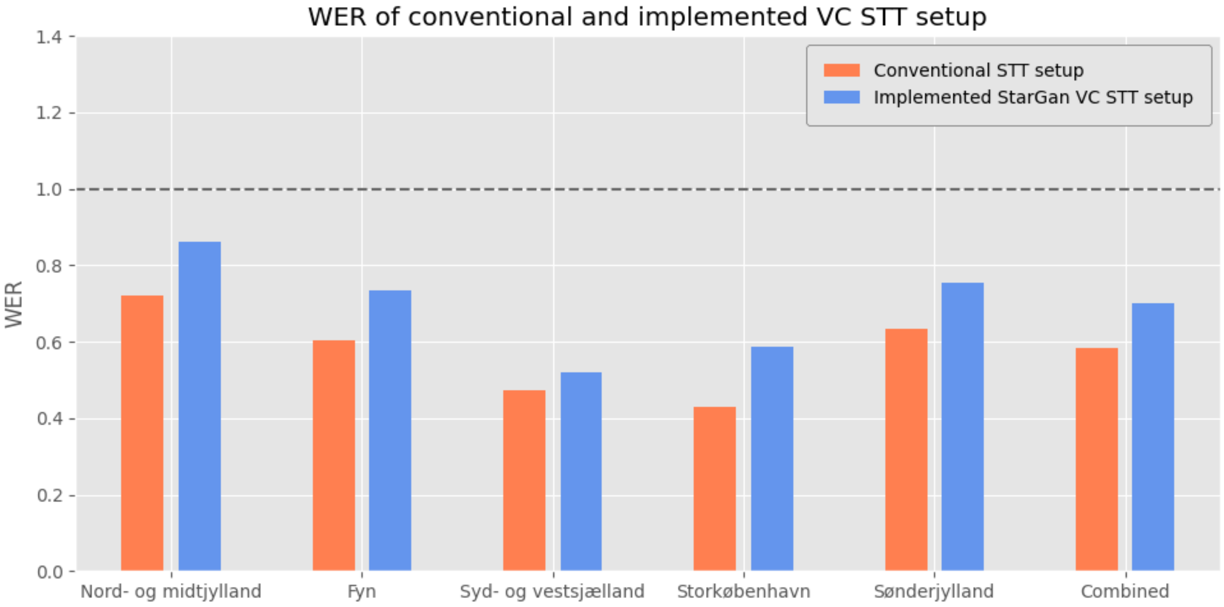


Figure 18: WER of danspeech transcriptions: One for a model retrained on original Spraakbanken data and no voice conversion(conventional STT setup) and for a model retrained on StarGAN voice converted speaker data and voice conversions of test speaker inputs before transcribing (StarGAN VC STT setup). The WER scores are categorized for every dialect included in the project.

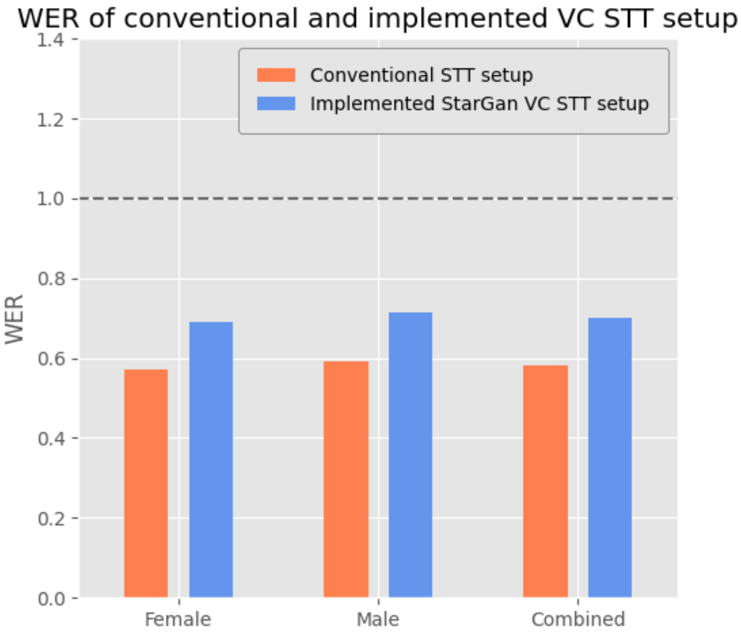


Figure 19: WER of danspeech for transcribing using conventional STT setup(retrained on Spraakbanken with no voice conversion) and StarGAN STT setup(retraining on voice converted Spraakbanken data and converting speaker input before transcribin). The scores are categorized by the speaker sex labels provided by Spraakbanken: male, female, and the two sexes together.

4.3.2 Results of Third Research Question

Looking at the WERs plotted in figure 18 the same tendencies exist as were the case in figure 16, with relation to which dialects perform well and which are worse. Here Syd- og Vestsjælland and Storkøbenhavn have particularly low (good) WERs compared to the other dialects. This is most likely due to the larger proportion of speaker in Spraakbanken with these dialects, as explained in the data section 2.2 figure 20.

Looking at the WER scores across sexes, figure 19 and table 2, the StarGAN voice converted STT setup have a slightly lower WER for females than males. However, the difference is small, and is very likely caused by the full Spraakbanken data set having a larger proportion of female speakers, as illustrated in appendix C.

	Nord- og midtjylland	**Fyn**	**Syd- og vestsjælland**	**Storkøbenhavn**	**Sønderjylland**	**Female**	**Male**	**Combined**
Original	0.72	0.604	0.474	0.43	0.633	0.573	0.593	0.582
StarGAN	0.863	0.735	0.521	0.587	0.755	0.692	0.713	0.701

Table 2: Word Error Rates for conventional STT and StarGAN VC STT setup using Spraakbanken *Training* Data. Measures are made across five dialects, female and male sex, and for the full combined speaker data.

One of the more interesting observations from the experiment is that the WERs of the StarGAN VC STT setup is well below 1. And, that the relative difference in performance for the conventional danspeech STT setup and the StarGAN voice converted STT setup is consistently around 0.05-0.15 across the different dialects and speaker sexes, which can be seen from the values in table 2. This indicates, that danspeech is capable of learning from the voice converted audio and that the conversions therefore do not remove all essential speech information need by danspeech. However the results clearly show that danspeech performs better on the original Spraakbanken *training* data.

5 Discussion

5.1 Discussion of Results

5.1.1 First Research Question

During investigation of the first research question, StarGAN and VAE were evaluated using a sanity check. The sanity check, which can be read in its entirety under section 4.1, showed that StarGAN was in fact able to produce claimed performance while VAE was not. Below, the performance of StarGAN, as well as VAE's lack of it, are discussed.

StarGAN Performance
StarGAN generally showed great performance, and made surprisingly good conversions, even on inter-lingual data.
During implementation of StarGAN, a pilot training on five male speakers was run, and pilot testing on this training showed that StarGAN was able to produce satisfying zero-shot conversions for female voices, which it had essentially never encountered before. This emphasizes the ability of StarGAN to essentially learn general mappings for its target speakers unrelated to the other speakers these were based upon. This is excellent for speaker style transfer in any-to-one zero-shot settings, as it is able to generalize so well beyond its training data.

The sanity check found noticeable volume fluctuations within different conversions, particularly ones that did not originate from Spraakbanken. Though audio-format and encoding for all speech samples included in conversions were identical, audio quality varied widely in non-Spraakbanken data both for better and worse, in contrast to Spraakbanken audio quality which while not being excellent was consistent.
In a StarGAN training run separately from the research question, speech data on a target speaker extracted from an audio book was included, as this allowed for a target speaker who had an abundance of seemingly high quality data. The conversions produced using this trained StarGAN model turned out to be not nearly as good as the only-Spraakbanken trained model, even though it was earlier found that more data for each speaker had a positive effect on conversion quality. The same way that fluctuations in audio volume occurred mainly in audio data with very different recording quality than Spraakbanken, the mere difference in recording quality, not lack of quality, seemingly interfered with StarGAN's ability to train properly. In this regard, it seems as if StarGAN is fragile when handling different qualities of recorded audio.

Why did VAE not Produce Satisfiable Conversions?
As presented in the VAE loss figure 15 under section 4.1.3, the KL loss and reconstruction loss point to a case of overfitting in the VAE model. The early drop in losses after only around 50,000 training iterations clearly doesn't fit with the 200,000 iterations made in the original VAE study. One likely reason for this might be the model hyper parameters. An idea could be to apply a larger dropout rate to the convolutions in the encoder, to avoid this overfitting, and make it capable of better encoding the new unseen source speakers. Also, the converted voice was dark and mumbling in a similar way for both female and male speaker conversions. This could indicate that a more general problem exists, like an error in the sampling rate or the saving of the .wav files.

Another reason for the poor voice conversion performance in VAE could stem from the choice of training data. The original VAE was trained on the English VCTK dataset with about 100 speakers with close to 400 utterances each. For the sanity check the VAE model was trained on approximately 40 speakers with close to 900 utterances each. If the number of training speakers had been increased, even with fewer utterances per speaker, this might have helped the VAE learn a more representative latent space of speaker representations, potentially resulting in better voice conversions.

5.1.2 Second Research Question

The findings of this test showed a number of aspects relating to the the performance of the VC models, danspeech and further the overall problem statement.

From the findings in 4.2 it was evident, that StarGAN obtained significantly better WERs than VAE. Together with the results from research question 1, section 4.1 , the findings support the idea that higher quality voice conversions in the sanity check lead to better speech to text translations in danspeech.
In the light of these findings, it was determined that VAE would not be utilized in retraining the danspeech STT model when investigating the third research question. This however does not mean that VAE could never be used as voice conversion model for the danspeech STT, but rather suggests that further adjustments and optimizations are needed in order for the implemented VAE to obtain higher quality audio output. If this is achieved, the model might still be a suitable choice.

It was clear that the pre-trained danspeech model performed significantly better on the original speaker data compared to converted data, even for StarGAN which produced acceptable voice conversions. These performance differences are attributed to the quality of the voice conversions, which while being good still were not as clear and high quality as the original speaker data. Hereby, the inclusion of the third research question is justified, as it is clear that danspeech needs to be accommodating for the lesser quality speech produced by StarGAN. Through retraining danspeech, the idea was that this accommodation would indeed happen.

The performance of danspeech was significantly better when transcribing speech by speakers with dialects from Syd- Vestsjælland and Storkøbenhavn. This supports the theory of danspeech being superior in transcribing speech uttered by speakers who are among the majority of Danish dialects, on which more training data is accessible.

A further point is that the test found a slightly better performance in the pretrained danspeech model when transcribing female speakers compared to male speakers. The difference could be attributed to an unnoticed deficit in audio quality of the male speaker data. But, it could also originate from having more training data on female speakers as seen in the data described in appendix C. This project did not get to investigating this further, as the WERs were only tested with voice conversions from a male target speaker, yet if the reason for the performance difference is due to the speaker sex, then it might be worth investigating the performance of danspeech when converting to a female target speaker.

5.1.3 Third Research Question

The results from investigating the third research question showed that the proposed VC-implemented STT setup did not improve the transcriptive performance of danspeech compared to the conventional setup. The reasons as to why the proposed setup did not improve performance can be attributed to a number of different factors, the most probable of these are discussed below.

Loss of Linguistic Information Quality
As touched upon in the discussion of the second research question, the quality of the speech samples provided to danspeech had a seemingly big influence on its transcriptive performance measured in the WER score. The discussion hinted that retraining the danspeech model would likely enable it to accommodate for this deficit in quality, but this turned out not to be the case. It is very likely that the danspeech STT model relies somewhat heavily on the quality of speech, and even though the performance of conversion by StarGAN has been evaluated to be acceptable, the audio quality of said conversion is noticeably lower than the original samples. This lack in audio quality is deemed to be directly affecting the performance of the STT training negatively.
A VC system able to produce better quality audio could potentially solve this problem, and it would be interesting to examine the danspeech system performance with respect to the quality of training data.

Lack of Normalization in VC
The general idea behind implementing VC into the STT setup is that VC can convert all speakers into a common speaker, effectively normalizing speech inputs on which danspeech has to train and transcribe.
Though all speakers in Spraakbanken resemble the chosen target speaker quite well after StarGAN conversion, there is still differences in pronunciations of words and sentences that giveaway the fact that they do not originate from the same person. To the human listener the converted Spraakbanken data set sounds much less varied, but it is possible that it still contains as much audiological variance as the original data set. If this is the case, the idea of neutralizing input data becomes more complex, and maybe VC is not the ideal approach to neutralizing speech. It could be interesting to make a more audio-focused investigation of the conversions, in which the goal would be to understand how much audiological variance the conversion process actually leaves when converting to a single target speaker voice.

Final Thoughts - danspeech Learning Capabilities on VC Data
As all results for the StarGAN STT setup research question 3 yielded WER scores well below 1, there are indications that danspeech is capable of learning speech to text mappings through voice converted data. Therefore it is possible that further improvements of the VC models will lead to better WER scores than the results this project has achieved. The question then becomes if the optimum for the improved VC models will be good enough to in fact increase the best WER scores of the existing danspeech models, or if the improvements made without VC will continuously supersede the VC implemented proposal.

5.2 Technological Discussion and Future Work: Improving Speech-to-Text Using VC Technology

5.2.1 Data Quality and Cleaning Strategies

The quality of training data, both for VC-models and STT-models, certainly has an impact on the performance of said models, and in this regard discussing possible options for improving data quality is of interest.

The data used in this project, described in detail under section 2, is generally of good quality but with exceptions that lead to potential for improvement. The Spraakbanken dataset includes numerous recordings containing abnormally long periods of silence, speakers repeating sentences in the same recording due to misspoken parts, background noise etc. Audio data, qualitatively damaged during the recording process, is naturally unavoidable as a product of human nature and having to record 300+ utterances in a single sitting. Nonetheless it interferes with training of most voice conversion models. Clever ways to remove bad recordings could help to significantly reduce the amount of human errors involved in recording large amounts of utterances, also filters could be applied to remove noise and other interference which would further improve the audio quality.

In regards to audio filtering, another interesting approach would be using convolutional autoencoders for denoising the speaker input, which has shown promising results in speech denoising [31]. Using such denoising models on the speaker input before the voice conversions are applied might lead to a clearer speaker conversion and thereby a better speech to text translation in the end.

The above discussion of data quality sheds light on an important aspect that this project, and voice technology in general, has to face: Obtaining good quality speech audio is hard and expensive, and whatever can be done to ease the process of obtaining it will most likely help improving most voice technologies.

StarGAN and danspeech also hold potential for improvements in regards to the data they utilize. Both these models operate on 16 kHz audio, described under section 2.4. This is generally sufficient for representing speech, but it is not able to represent the entire audible spectrum which requires a minimum of 40 kHz. This creates a potential deficit in information between the actual recorded speech, and the down-sampled audio that is utilized as data by the models. The main challenge with using higher sample-frequency audio is the increased computational load and amount of data that it naturally leads to. This would in some cases make building the models infeasible due to time and memory limitations during data preprocessing and model training, which is why it may be advantageous to choose a lower sampling rate like 16 kHz used in this project.

5.2.2 Expanding Model with Dialect Specific Conversion

The strategy of converting speakers to a common speaker style, as investigated by this project, was not sufficient in terms of how much the speakers acoustic features were neutralized. Instead of changing the general speaker style of source speakers, focusing more specifically on changing the dialect-defining acoustic features may be of interest. Existing studies of dialect transformation have shown some promising results that could be worth investigating further [15]. Potentially, adding a model specifically for dialect transformations together with a model for speaker style transformations might create a common voice that is more similar across the different dialects than the one achieved in this project. This could potentially improve the final speech recognition and word error rates.

5.2.3 Further Optimization of the Combination of VC and STT Models

This project set out to investigate if voice conversions could be utilized for improving speech to text translation with danspeech. For that reason, the models from StarGAN [13] and VAE [23] were used under the assumption that model configurations made in their studies were sufficiently optimized for the evaluation with danspeech. Writing this project, the focus was not on the individual voice conversion and speech to text models, but on their combined performance. In future work it would be interesting to experiment with model configuration and hyper parameter tuning, both for the systems separately, but further for the combined VC-STT model system as a whole.

In such combined case, the full hyper parameter space would be quite large and the training times would be significantly increased as each iteration would require a training of the voice conversion model, a conversion of speaker data, followed by a retraining of danspeech and a word error rate test to use as the performance measure of the overall setup. For that, Bayesian Optimization would be a fitting choice to avoid searching the whole hyper parameter space.

5.2.4 Voice Conversion as Data Augmentation

The strategy of retraining an STT model on one single speaker voice by converting all remaining speakers to that voice has been presented in this project. An interesting alternative, which could possibly be investigated in future studies, would involve utilizing this VC implementation as a data augmentation tool. Instead of converting all speaker training data into only one common voice, the alternative approach would be to convert all speaker training data into all of the speakers voices, essentially multiplying the amount of data with the number of unique speakers. In terms of feasibility, converting all speaker data to all voices is straight forward with many-to-many voice conversion, but the amount of data will require optimization in regards to memory load and running times on the massive data sets which would be produced. An example, is StarGAN conversions of Spraakbanken used for retraining danspeech, which were 49 gigabytes of converted audio data. Creating the augmented dataset would involve making 616 speakers worth of conversion sets amounting to around 30.184 terabytes of audio data, quite a substantial amount.

5.3 Ethical Considerations of Using VC Technologies

The utilization of voice conversion in speech recognition technologies, and communication between individuals in general, poses both potential benefits and drawbacks. In this section, the societal and ethical considerations related to the impact of implementing VC technologies are discussed.

5.3.1 A Game of Cat and Mouse

Looking at the current landscape of voice conversion technologies, great advances have been made in the discipline of generative models that can synthesize voices almost indistinguishable from real voices, as demonstrated in this project. However, the development of technologies that are able to detect these generated fake voices, and deep fakes in general, is lagging behind [42]. This poses a challenge and a potential threat to all of society, as deep fake detection technologies are essential for preventing their misuse. Examples of the criminal misuse of VC already exist, like fraudsters impersonating a CEO to steal $243,000 [27], but it is not only money that is on the line. With the ability to convert voices, it becomes possible to put words in other peoples' mouths [35], and this poses a threat to the information society the civilized world is build upon. Good VC has the potential to completely social media, news and politics, and in some cases properly is already. As the VC technology improves, more and bigger cases are likely to emerge in which VC was used maliciously, and while some of these may be detected, other malicious uses of VC may fly so far below the radar and be advanced enough that nobody will be able to detect that what they encountered was actually fake. In order to be able to prevent the misuse of VC and other deep fake technologies, the advancements in these fields has to continuously be developed for good, and it is of great importance that any inventor of voice conversion technologies considers the potential risks they might create through their invention.

5.3.2 The Ease of Use for Voice Conversion Technologies

For VC technologies to have substantial impacts to society, both positive and negative, they need to be easy and cheap to deploy. Current state-of-the-art voice conversion models, such as StarGAN and VAE examined in this project or the popular WaveNet [10], though being implementable by many still require lots of high quality speech data and computing power to function well. Here knowledge of deep neural networks, hyper parameter tuning, and the specific model architectures are further necessary, like in the case of this project. Fitting these models to particular people, dialects or languages still requires a bunch of know-how to make work, but might soon be simple plug and play systems. As such, the immediate potential for good uses as well as threatening ones of VC deep fakes could be quite low, but will increase with the maturation of the technology.

5.3.3 The Motivation for Better Speech to Text - Bridging the Minority Gap

This project seeks to utilize VC for good, and specifically further bridge the gap between people speaking the same language, but different dialects. Some dialects will continuously become more rare, like the dialect spoken by some in Sønderjylland which holds a population of around 250,000 citizens. Meanwhile, other dialects of speech will become the vast majority, an example might be the ones spoken in almost the entire region of Sjælland and the capitol area that together has an estimated population of 2,300,000, almost ten times larger than Sønderjylland [1].

This tendency of language convergence might result in less represented dialects such as the one of Sønderjylland becoming actual dialect minorities, for which particular services and tools are not available. This is an ethical dilemma which should be addressed. On one side, if a utilitarian approach was used, one might argue that the action producing the largest amount of wealth should be taken. In this case, that might be to produce services and tools for the region of dialects from Sjælland, as they clearly make up the largest amount of people, and will increase as the Danish dialects further converge to dialects from and around Copenhagen [1]. On the other hand, a deontological approach might be more suitable in this case. It could be argued that the people of Sønderjylland are citizens with a right to receive the same quality of services and tools as the rest of the Danish citizens. Therefore it is a societal duty to make sure that these services and tools are fitted to the needs of the minor group of people as well.

Besides the ethical considerations regarding applications of voice conversion technologies in a societal perspective, the prospect of uniting people across dialects may be a more pragmatic way of seeing the matter. As VC technologies become increasingly better, removing the spoken barriers that, to varying extends, divide different population groups of Denmark and other countries will become easier to break down. If VC implementation in different government provided services could help bridge the dialect minority gap by supporting all dialects, it might bring with it increased equality and other positive effects.

Attempting to bridge the gap between dialects should be done carefully though. Dialects are tied closely to cultural identities of its people [4], which the use of VC effectively seeks to diminish. In this regard, the question becomes if voice conversion technologies for dialects could potentially diminish this cultural difference between groups of a diverse population and if this would be damaging to the cultural diversity of a country. This and other implementations pose possibly interesting studies to be conducted after the VC technologies have been applied on a societal scale.

For this particular project, the speakers style and dialectic identity is only affected inside the STT system and not passed on to human listeners. As such, implementing this project in society would most likely have a lesser risk of reducing cultural diversity.

5.3.4 Safe AI - Considerations for Project Applications

If this project was to be applied in a real world context, it would be wise to consider ethics as part of the implementation process. As a means of accommodating potential risks and challenges, this project therefore proposes to follow relevant guidelines from *Safe AI Principals*[20]. In particular the following principles are deemed important for this project:

- **Failure Transparency** - If the system fails and causes problems or harm, it must be possible to understand where and why, the failure happened. In other words, the implementation of the system must be build around explainability.
- **Responsibility** - Designers and builders should be seen as stakeholders in the use, and misuse of the system. This should enforce ethical thinking among the team implementing the system.
- **Personal Privacy** - The users should always have access to and full control over their data. Further, the system should be implemented using privacy by design to reduce the risk of misuse from third parties.
- **Shared Benefit** - Finally, the implementation and use of the system should be done as to support and empower as many people as possible, and not just cater to the biggest user segment or the largest profits.

It will never be possible to extinguish all risks and ethical challenges, but following these principles would be a way of reducing them.

6 Conclusion

During the investigations conducted in this project, a number of insights were found.

Firstly, evaluated on the basis of a sanity check, StarGAN proved that it was possible to implement state-of-the-art VC technology and produce good sounding voice conversions in an any-to-one zero-shot setting. This was not the case for VAE, which failed in being able to reproduce claimed state-of-the-art results despite of a heavy investment in getting it to work. However, the insufficient voice conversions are not necessarily caused by the VAE technology in general, but more likely because the implemented VAE needed to be further fine-tuned.

The testing conducted under the second research question enforced the notion that the implemented StarGAN was much better than VAE, yet also showed that danspeech performed reasonably better on original speech compared to the StarGAN converted speech. It was argued that the difference in performance was likely in part due to the better audio-quality of original speech samples, which was believed to heavily influence STT performance, but that retraining the models as of the third research question would potentially accommodate for this difference. On the basis of these results, VAE was scrapped and focus was moved to investigate the performance of danspeech retrained using StarGAN as the VC model.

Testing the two retrained danspeech models; one trained on original speech transcribing original voices and the other trained on converted speech transcribing only voices converted to a single target voice, found that the original STT setup was superior to the VC implemented setup. Despite of this, the results of this research question saw a decrease in performance difference between original and converted speech compared to the results of the second research question. The project thereby does not reject the possibility of voice conversion setups being a viable option for improving the performance of danspeech STT.
The project argues that producing convincing voice conversions is seemingly only half the story when trying to implement VC into a STT setup, and proposes that future work should focus on advancements in audio-quality of produced voice conversion. Preferably in combination with specific VC model adjustments to capture acoustic features specifically related to speaker dialects.

The authors of this project finally discuss ethical considerations related to the implementation of VC technology in society, and argue that this has to be done carefully while considering its potential impact and further seeking to eliminate missuses of the technology. As part of the ethical considerations, a set of general *Safe AI Principles* are suggested as guide lines when implementing VC technologies.

References

[1] Tore Kristiansen. “The Role of Standard Ideology in the Disappearance of the Traditional Danish Dialects”. In: *Folia Linguistica* 32.1-2 (1998). ISSN: 0165-4004. DOI: 10.1515/flin.1998.32.1-2.115. URL: https://www.degruyter.com/view/j/flin.1998.32.issue-1-2/flin.1998.32.1-2.115/flin.1998.32.1-2.115.xml.

[2] Tomoki Toda, Yamato Ohtani, and Kiyohiro Shikano. “One-to-Many and Many-to-One Voice Conversion Based on Eigenvoices”. In: *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP ’07*. Vol. 4. IEEE, Apr. 2007, pp. IV-1249-IV–1252. ISBN: 1-4244-0727-3. DOI: 10.1109/ICASSP.2007.367303. URL: https://ieeexplore.ieee.org/document/4218334/.

[3] Masanori Morise, Hideki Kawahara, and Haruhiro Katayose. “Fast and reliable F0 estimation method based on the period extraction of vocal fold vibration of singing voice and speech”. In: *Proceedings of the AES International Conference*. 2009.

[4] Altugan Arzu and Tozun Issa. “An Effect on Cultural Identity: Dialect”. In: *Procedia - Social and Behavioral Sciences* 143 (Aug. 2014), pp. 555–562. ISSN: 18770428. DOI: 10.1016/j.sbspro.2014.07.435.

[5] Ian J. Goodfellow et al. “Generative Adversarial Networks”. In: (June 2014). URL: http://arxiv.org/abs/1406.2661.

[6] Dario Amodei. “Deep Speech 2: End-to-End Speech Recognition in English and Mandarin”. In: (2015). URL: https://findit.dtu.dk/en/catalog/2289647626.

[7] Malene Monka. “Place and dialect levelling in Denmark”. In: *Locating Language* (2015). URL: https://curis.ku.dk/ws/files/157953468/Abstract_Ohio_Place_and_dialect_leveling_in_Denmark_pdf.pdf.

[8] Masanori Morise. “CheapTrick, a spectral envelope estimator for high-quality speech synthesis”. In: *Speech Communication* 67 (Mar. 2015), pp. 1–7. ISSN: 01676393. DOI: 10.1016/j.specom.2014.09.003.

[9] Masanori MORISE, Fumiya YOKOMORI, and Kenji OZAWA. “WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications”. In: *IEICE Transactions on Information and Systems* E99.D.7 (2016), pp. 1877–1884. ISSN: 0916-8532. DOI: 10.1587/transinf.2015EDP7457. URL: https://www.jstage.jst.go.jp/article/transinf/E99.D/7/E99.D_2015EDP7457/_article.

[10] Aaron van den Oord. “WaveNet: A Generative Model for Raw Audio”. In: (2016). URL: https://arxiv.org/abs/1609.03499.

[11] Takuhiro Kaneko and Hirokazu Kameoka. “Parallel-Data-Free Voice Conversion Using Cycle-Consistent Adversarial Networks”. In: (Nov. 2017). URL: https://arxiv.org/abs/1711.11293.

[12] Darshana Buddhika et al. “Voicer: A Crowd Sourcing Tool for Speech Data Collection”. In: *2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer)*. IEEE, Sept. 2018, pp. 174–181. ISBN: 978-1-5386-7352-2. DOI: 10.1109/ICTER.2018.8615521. URL: https://ieeexplore.ieee.org/document/8615521/.

[13] Hirokazu Kameoka et al. “StarGAN-VC: Non-parallel many-to-many voice conversion with star generative adversarial networks”. In: (June 2018). URL: https://arxiv.org/abs/1806.02169.

[14] Songxiang Liu et al. “Voice Conversion Across Arbitrary Speakers Based on a Single Target-Speaker Utterance”. In: *Interspeech 2018*. Vol. 2018-. ISCA: ISCA, Sept. 2018, pp. 496–500. ISBN: 9781510872219. DOI: 10.21437/Interspeech.2018-1504. URL: http://www.isca-speech.org/archive/Interspeech_2018/abstracts/1504.html.

[15] Nath Sanghamitra and Sharma Utpal. *Incorporating Dialectal Features in Synthesized Speech using Voice Conversion Techniques*. Tech. rep. 19. 2018, pp. 975–8887. URL: http://www.lmp.ucla.edu/Profile.aspx?menu=004&LangID=83.

[16] Thilo Spinner et al. *Towards an Interpretable Latent Space – An Intuitive Comparison of Autoencoders with Variational Autoencoders*. Oct. 2018.

[17] Ju-chieh Chou, Cheng-chieh Yeh, and Hung-yi Lee. “One-shot Voice Conversion by Separating Speaker and Content Representations with Instance Normalization”. In: (Apr. 2019). URL: http://arxiv.org/abs/1904.05742.

[18] Orhan Ocal. “Adversarially Trained Autoencoders for Parallel-Data-Free Voice Conversion”. In: (2019). URL: https://findit.dtu.dk/en/catalog/2447101689.

[19] *12. Virtual Environments and Packages — Python 3.8.3 documentation*. URL: https://docs.python.org/3/tutorial/venv.html.

[20] *AI Principles - Future of Life Institute*. URL: https://futureoflife.org/ai-principles/?cn-reloaded=1.

[21] *An Intuitive Explanation of Connectionist Temporal Classification*. URL: https://towardsdatascience.com/intuitively-understanding-connectionist-temporal-classification-3797e43a86c.

[22] Cambridge. *DIALECT — meaning in the Cambridge English Dictionary*. URL: https://dictionary.cambridge.org/dictionary/english/dialect.

[23] Ju-Chieh Chou, Cheng-Chieh Yeh, and Hung-Yi Lee. *One-shot Voice Conversion by Separating Speaker and Content Representations with Instance Normalization*. Tech. rep. URL: https://github.com/jjery2243542/.

[24] *CSTR VCTK Corpus*. URL: https://homepages.inf.ed.ac.uk/jyamagis/page3/page58/page58.html.

[25] danspeech. *danspeech · GitHub*. URL: https://github.com/danspeech.

[26] *DanSpeech’s Documentation — DanSpeech 1.0.0 documentation*. URL: https://danspeech.github.io/danspeech/html/index.html.

[27] *Fraudsters Used AI to Mimic CEO’s Voice in Unusual Cybercrime Case - WSJ*. URL: https://www.wsj.com/articles/fraudsters-use-ai-to-mimic-ceos-voice-in-unusual-cybercrime-case-11567157402.

[28] Xun Huang and Serge Belongie. *Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization*. Tech. rep.

[29] *Intuitive Guide to Understanding KL Divergence - Towards Data Science*. URL: https://towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-understanding-kl-divergence-2b382ca2b2a8.

[30] *jiwer · PyPI*. URL: https://pypi.org/project/jiwer/.

[31] Mike Kayser and Victor Zhong. *Denoising Convolutional Autoencoders for Noisy Speech Recognition*. Tech. rep.

[32] *McNemar's Test - mlxtend*. URL: http://rasbt.github.io/mlxtend/user_guide/evaluate/mcnemar/.

[33] *Mean Opinion Score - an overview — ScienceDirect Topics*. URL: https://www.sciencedirect.com/topics/engineering/mean-opinion-score.

[34] Mozilla. *Common Voice*. URL: https://voice.mozilla.org/en/about.

[35] *New Deepfake Method Can Put Words In Anyone's Mouth - VICE*. URL: https://www.vice.com/en_us/article/g5xvk7/researchers-created-a-way-to-make-realistic-deepfakes-from-audio-clips.

[36] *NST Dansk ATG-database (16 kHz) – Språkbanken*. URL: https://www.nb.no/sprakbanken/ressurskatalog/oai-nb-no-sbr-19/.

[37] *PyTorch*. URL: https://pytorch.org/.

[38] PyTorch. *PyTorch*. URL: https://pytorch.org/.

[39] *pyworld · PyPI*. URL: https://pypi.org/project/pyworld/.

[40] *torch.utils.data — PyTorch master documentation*. URL: https://pytorch.org/docs/stable/data.html.

[41] *Variational autoencoders*. URL: https://www.jeremyjordan.me/variational-autoencoders/.

[42] Run Wang et al. *DeepSonar: Towards Effective and Robust Detection of AI-Synthesized Fake Voices*. Tech. rep. URL: https://arxiv.org/pdf/2005.13770.pdf.

[43] *WaveNet: A generative model for raw audio — DeepMind*. URL: https://deepmind.com/blog/article/wavenet-generative-model-raw-audio.

[44] *What is Word Error Rate (WER)?* URL: https://blog.deepgram.com/what-is-word-error-rate/.

Appendix

A List of Training Speakers Used in Sanity Check

This is the list of speakers used for training both StarGAN and VAE for the sanity check described in section 3.7.5.

```
speakers = ['r6110005',  # Oestjylland F
            'r6110008',  # Vestjylland M
            'r6110009',  # Oestjylland M
            'r6110011',  # Nordjylland M
            'r6110018',  # Fyn F
            'r6110019',  # Fyn M
            'r6110022',  # Vestjylland F
            'r6110023',  # Oestjylland M
            'r6110026',  # Vestjylland M
            'r6110027',  # Storkoebenhavn M
            'r6110028',  # VestSydSjaelland F
            'r6110030',  # Storkoebenhavn M
            'r6110031',  # Nordjylland F
            'r6110034',  # Fyn M
            'r6110037',  # VestSydSjaelland F
            'r6110038',  # Storkoebenhavn F
            'r6110041',  # Soenderjylland F
            'r6110042',  # VestSydSjaelland M
            'r6110043',  # Oestjylland F
            'r6110044',  # VestSydSjaelland M
            'r6110046',  # VestSydSjaelland M
            'r6110048',  # Storkoebenhavn F
            'r6110049',  # Soenderjylland F
            'r6110050',  # Target Storkoebenhavn M
            'r6110051']  # Fyn F
```

B List of Symbols Accepted by danspeech for Retraining

This list shows all symbols that danspeech allows for retraining their models. This means all symbols, that are not in the list should be removed before retraining danspeech.

[
”_”, ”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”, ”æ”, ”ø”, ”å”, ”é”, ”ü”, ”ö”, ” ”
]

C Plot of Speaker Gender Distribution in Spraakbanken

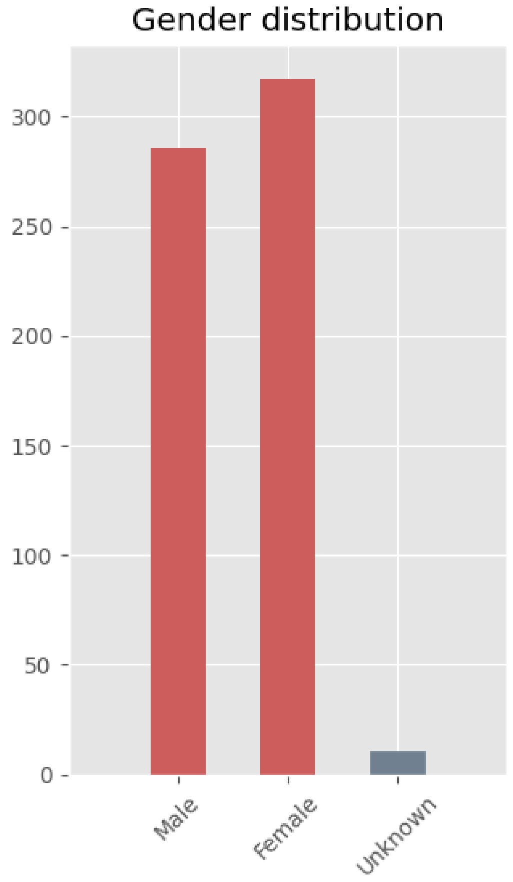


Figure 20: The gender distribution over the 614 speakers from Spraakbanken-train

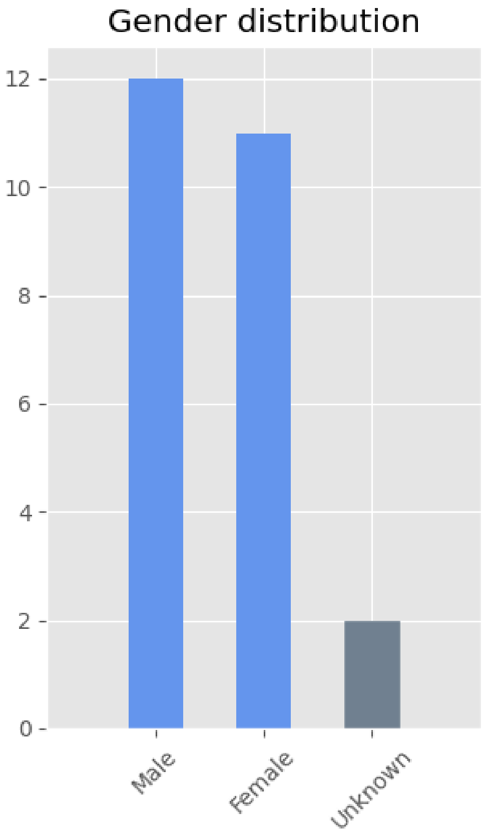


Figure 21: The gender distribution over the 25 speakers from Spraakbanken-test