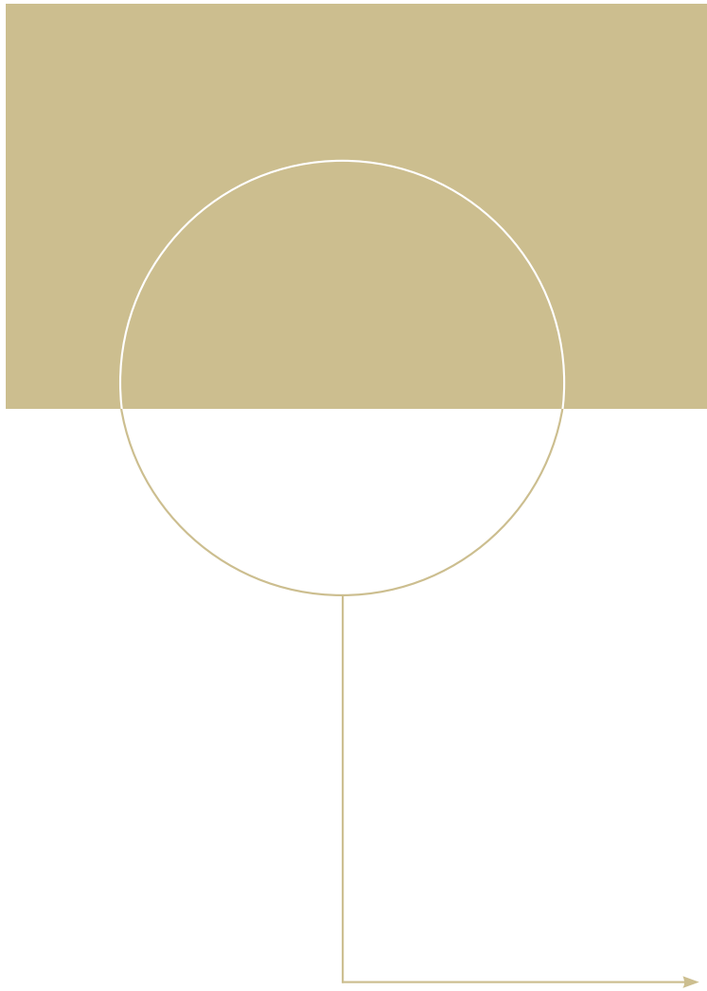


Master's thesis

2022

Diepiriye Deinma Okujagu

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology and Electrical
Engineering
Department of Electronic Systems

Diepiriye Deinma Okujagu

MASTER THESIS: VOICE CONVERSION

Trondheim, January 2022

January 2022

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

MASTER THESIS: VOICE CONVERSION

Trondheim, January 2022

Diepiriye Deinma Okujagu

Electronics Systems Design
Submission date: January 2022
Supervisor: Torbjørn Karl Svendsen
Co-supervisor: None

Norwegian University of Science and Technology
Department of Electronic Systems

ABSTRACT

Voice conversion (VC) is the process of altering one speaker's words to appear as though a different person is speaking them. The speech signal from the first speaker, known as the source speaker, should retain its linguistic content in its final converted form, known as the resulting voice. Still, it should also be maximally altered in terms of vocal timbre, range, inflection, and so on to match the voice of the second speaker, known as the target speaker. This activity can synthesize voices in various ways, and it could be a vital component in creating human-sounding artificial voices for robots.

Using a form of a generative adversarial network (GAN) called StarGANv2 originally developed by Yinghao Li et al.., this project provides an approach that permits non-parallel many to many and cross-gender voice conversion (VC). The model used is called StarGANv2-VC. It is unique in that it (1) does not require parallel utterances, transcriptions, or temporal alignment processes for speech generator training, and (2) learns many-to-many mappings across several attribute domains using a single generator network. Our approach greatly outperforms earlier VC models using a combination of adversarial source classifier loss and perceptual loss. This paradigm applies to a wide range of speech conversion jobs, including many-to-many, cross-lingual, and singing voice conversion.

ACKNOWLEDGEMENT

I want to acknowledge the Almighty God for giving me the grace, knowledge, wisdom, and ability to carry out this project. Also, my parents, Uncle Muyeed, who unfortunately was taken by Covid-19, and my sister, Lois, for all their encouragement, moral and financial support. Special thanks to my supervisor, Torbjorn Svendsen, and Zijian Fan for their help while working on this master.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

ASR-Automatic Speech Recognition

ANN=Artificial Neural Networks

CER=Character Error Rate

CTC-Connectionist Temporal Classification

CycleGAN=Cycle-consistent Generative Adversarial Network.

GAN= Generative Adversarial Networks

GMM= Gaussian Mixture Models

LPCC=Liner Predictive Cepstral Coefficients

MCEP=Mel Cepstral Coefficients

MFCC=Mel-Frequency Cepstral Coefficients

NLP= Natural Language Processing

PSOLA=Pitch Synchronous Over-Lap and Add

STRAIGHT=Speech Transformation and Representation using Adaptive Interpolation of Weighted Spectrum

TTS=Text-to-Speech

VC = Voice Conversion

WER=Word Error Rate

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND

The purpose of this project is to create a standalone voice conversion application that uses Deep Learning, a non-parallel data modeling method, in conjunction with a Star GAN network that emphasizes adversarial classifier loss, use of style encoders, F0(fundamental frequency) consistency, as well as encoders and decoders.

Voice conversion is a technique for converting a source speaker's voice to a target speaker's voice by keeping some features of the input voice, such as the content of the speech, while modifying other aspects, such as style and frequency. Text-to-speech (TTS), speaking assistance, speech improvement, pronunciation conversion, and other activities can all be accomplished with VC.

The Gaussian mixture model (GMM) is one of the most widely used techniques today, and it is based on statistical models. Non-negative matrix factorization (NMF), neural networks (NNs), restricted Boltzmann machines (RBMs), and deep learning are examples of other standard statistical models used in speech conversion.

The VC system is given a collection of utterances recorded from the source and target speakers during the training phase (the training utterances). The speech waveform signal is encoded into a representation that allows speech attributes to be modified during the speech analysis and mapping feature calculation steps.

Following the mapping features from a new source speaker utterance, the features are transformed using the trained conversion function in the conversion phase. From the converted features, speech features are computed, then used to synthesize the converted utterance waveform. VC approaches can be classified in a variety of ways. Whether they require parallel or non-parallel recordings throughout their training period is one thing to consider. Parallel recordings are utterances with identical linguistic content that differ only in the feature that must be mapped (speaker identity, in the VC case) (Mouchtaris et al., 2004). Another element to consider is whether they are text-dependent or text-independent (Ney et al., 2004). Word or

phonetic transcriptions, as well as the recordings, are required in text-based techniques. Parallel sentences from both the source and target speakers may be necessary for these methods. Because no transcription is available for text-independent techniques, these approaches must first identify speech segments with similar content before constructing a conversion function(Sundermann, 2008). The language used by the source and target speakers is a third aspect to consider. The assumption behind language-independent or cross-language VC is that the source and destination speakers speak different languages (Sundermann et al., 2003; Türk, 2007)

Many of the tasks in natural language processing (NLP) include constructing robots that can do jobs similar to those performed by humans. We consider language to be profoundly personal and cultural at the same time. In this regard, the task of a machine speech generator is one of the most astounding NLP techniques - it is essentially a machine speaking like a human. Create a device that not only talks like a human but speaks like a specific human to give an extra layer of impressiveness. Given a clip of a human voice, use a machine to make vocal samples of the same human voice saying different things. This refers explicitly to many-to-many voice conversion, in which the model may learn several voices and transfer the speech style between them in any combination.

This report will introduce a many-to-many non-parallel voice conversion model based on a variant of a Generative Adversarial Network (GAN) called StarGANv2-VC, which requires no parallel utterances transcriptions or time alignment procedures. This area has seen a lot of activity in recent years. Practically all of the studies on which we based our research were published within the previous year. The background research and related work that went into this publication will be described next.

Problem Formulation

This project aims to convert one person's (source) voice into another person's (target) say while retaining linguistic information.

The two main questions about a voice conversion system are 1) "How natural does the

converted voice sound?" and 2) "How similar does the converted voice sound to the target voice and the source voice?". Earlier Statistical methods and use of parallel utterances, including Deep Learning problems, have been used to solve this problem which also had limitations:

- The majority of voice conversion models require a large amount of parallel data of source-target speakers for model training, which is the main disadvantage because collecting pairs of utterances of the same sentences spoken by the source and target speakers requires a significant amount of effort.
- If there is a considerable acoustic gap between both utterances, it is difficult to align both source and target speech data using parallel data. It is necessary to make manual corrections for the alignment to work reliably, but it is not guaranteed to be completely aligned.

These are the few among others; hence, using a generative adversarial network (GAN) named StarGANv2, we describe an unsupervised non-parallel many-to-many voice conversion (VC) technique. Our approach beats earlier VC models using a combination of adversarial source classifier loss and perceptual loss and produces natural-sounding speech in terms of naturalness and speaker similarity.

1.2 OBJECTIVES

We use the StarGANv2 architecture for Voice conversion as introduced by Li et al (Li et al., 2021) which is a newly developed GAN architecture for picture style transfer, a new technique for unsupervised nonparallel many-to-many cross-gender voice conversion, which would be achieved by:

- Applying StarGANv2 to voice conversion, which enables converting from plain speech into speech with a diversity of styles
- Introducing a novel adversarial source classifier loss that significantly improves the similarity in terms of speaker identity between the converted speech and target speech Application of neural network models for velocity inversion on previously unseen seismic observations.
- Learning from existing data instead of using physical models and formulas.

1.3 LIMITATIONS

The objective of this work was initially aimed at cross-gender and bi-lingual voice conversion using Nigerian Pidgin English and British English Language. Still, unfortunately, due to the lack of speech data available for the Nigerian Pidgin English, we had to stick to just the British English open for this work.

CHAPTER 2

THEORETICAL BACKGROUND

This chapter discusses the theory of VC. Reading this should provide sufficient background information on the subject to enable the reader to follow the remainder of the thesis. To conduct VC, we must first encode speech mathematically in a way that allows us to alter the signal's properties, as voice conversion is a subset of speech synthesis.

The advancements in the fields of artificial intelligence (AI), machine learning (ML), and deep learning (DL) have widened the scope of these study domains' applications. During the early stages of evolution, applications were restricted to a narrow range of fields, such as biometrics, image processing, and pattern recognition. However, subsequent improvements covered novel application areas on the outskirts of exploration, such as digital speech processing (DSP), natural language processing (NLP), and electroencephalography (EEG) signal processing. Although, in recent years, sub-domains of DSP have emerged as a critical topic of research for deep learning researchers, including voice recognition, speaker recognition, speech synthesis, and speech augmentation. Among these, voice synthesis is a promising area of research that involves the artificial creation of human speech. (Dhar, 2021)

The purpose of voice conversion is to alter the features of a source speaker's voice using signal processing techniques so that the output can be identified as the voice of a target speaker. During the early years of VC research, this topic was extensively researched utilizing statistical algorithms such as the Gaussian mixture model (GMM), dynamic kernel partial least squares (DKPLS), and non-negative matrix factorization (NMF), among others. However, as the field of deep learning has advanced in recent years, VC has emerged as an essential topic of research for the DL community. The process of VC involves changing speech parameters such as fundamental frequencies (F0), spectral envelope, and formant structures to provide a natural-sounding speech. VC-based technologies are widely used in various real-time applications, including audio assistance devices for people with speech disorders, voice-over for film dubbing, and speech-to-singing conversion (Dhar, 2021). It generally employs two stages: training and transformation. During the training step, the voice conversion system receives information from the source and targets the speaker's voices, and automatically generates voice conversion rules.

To do this, training databases from source and target speakers are evaluated acoustically, and a mapping between the two speakers' acoustic regions is estimated. The transformation stage uses the mapping established during the training stage to change the source voice signal to fit the target voice's characteristics. The alteration is accomplished by applying a suite of signal processing techniques that alter the vocal tract and prosody features. In other words, a voice conversion system adjusts only the features of speech that are reliant on the speaker, such as formants, fundamental frequency (F0), intonation, intensity, and duration, while retaining the speaker-independent speech content. (Mohammadi and Kain, 2017).

A typical framework for voice conversion consists of three steps:

1) analysis of speech,
2) feature mapping, and
3) reconstruction of speech

which we refer to as the analysis-mapping-reconstruction pipeline.

Section 2.1 provides an overview of how speech works and how it is analyzed, followed by the feature mapping and reconstruction. When we understand how speech is physically produced, we can see how this information leads to the approach of voice conversion.

2.1 SPEECH PRODUCTION

Speech is a fundamentally human mode of communication. The development of computing systems capable of processing speech in a variety of ways is an exciting and critical task. Speech recognition and text-to-speech systems, for example, have garnered extensive interest due to their critical uses in enabling accessibility for disabled users, as well as in human-computer interface design and security systems. Specific systems, such as speaker identification systems, are primarily concerned with the timbral quality of speech, while others, such as singing voice synthesis, are equally concerned with intelligibility and naturalness.

It is necessary to consult the three forms of information conveyed by human speech:

- segmental data (relative to the quality of the voice);
- supra-segmental data (related to prosody),
- linguistic data (expressed by the series of phonemes uttered).

When a single utterance is considered, voice quality might be regarded as stable or slowly changing, whereas phonemes very fast over time. To collect all of the necessary information from a time-varying voice signal, it is required to divide it into short segments (frames) that can be considered stationary in themselves. The shorter a frame is, the more stationary its contents become.

Nonetheless, there is a natural limit to the frame size imposed by the third sort of speech information, prosody.

To do voice conversion, speech analysis/synthesis is required. The objective is to extract speech elements that allow for a significant change of speech's acoustic qualities. This is seen in Figure 2.1.

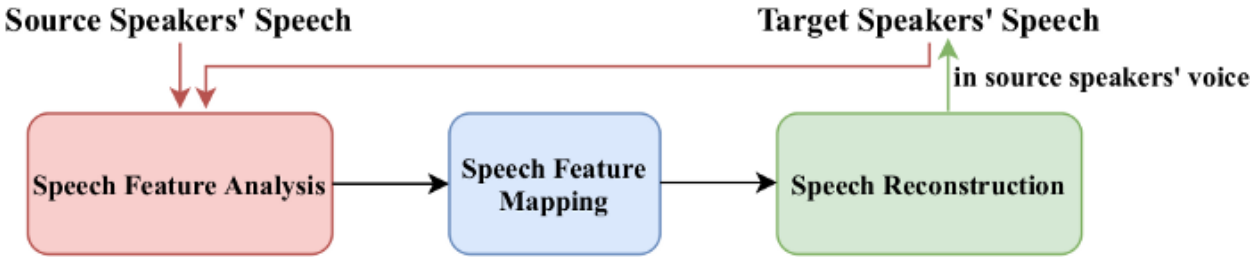


Figure 2. 1: Training and conversion phase of a typical voice conversion system (Dhar, 2021)

2.1.1 SPEECH ANALYSIS

The purpose of speech analysis is to deconstruct speech signals into some intermediate representation that may be used to manipulate or modify speech's acoustic features effectively. Numerous beneficial intermediate representation strategies have been investigated initially for speech communication and speech synthesis. They are pretty helpful for voice conversion. The approaches can be broadly classified into Signal-Based representations and Model-Based Representations.

i. **Signal-Based representations**:

Signal-based analysis/synthesis techniques mimic the speech signal without making limiting assumptions (such as the independence of the source signal and filter); as a result, they typically produce higher-quality results. The disadvantage is that they are less adaptable to change (Mohammadi and Kain, 2017). Signal-based representation techniques include Pitch Synchronous Over-Lap and Add (PSOLA). It divides a speech signal into overlapping speech segments, each representing one of the speech signal's subsequent pitch periods. We can reconstitute the voice signal of a particular intonation by overlapping and combining these speech segments with varied pitch durations.

PSOLA's analysis and reconstruction do not introduce substantial artifacts because it works directly on the time domain speech signal. While the PSOLA technique is excellent for changing the fundamental frequency of voice sounds, it has significant drawbacks. Unvoiced speech signals, for example, are not periodic, making time-domain signal manipulation difficult.(Moulines and Charpentier, 1990)

Also, HNMs (harmonic plus noise models) assume that the speech signal can be broken down into harmonics (sinusoids with frequencies relevant to pitch). HNMs produce high-quality speech, but they are less adaptable for adjustment than source-filter models, owing to the complexity of dealing with phase. (Stylianou, 1996)

ii. Model-Based Representations

The model-based technique is based on the assumption that a model with time-varying parameters may mathematically represent the input signal. The source-filter model is a common illustration. Because source-filter models are more adaptable, VC may be advantageous. The source-filter model is used in this thesis. Speech is seen as a mix of an excitation signal (or source signal) and a filter when utilizing a source-filter model (Mohammadi and Kain, 2017). The filter symbolizes the vocal tract, while the source signal depicts air coming out of the vocal cords. Because the source signal and the filter are regarded as separate entities, we can adjust them separately. The filter is viewed as a time-varying filter since the vocal tract is continually changing with the position of the articulators.

Either a voiced or an unvoiced excitation signal can be used. An impulse train can be used to mimic a voiced excitation signal, with the space between the pulses varying with

the speaker's fundamental frequency. On the other hand, the unvoiced signal can be described as noise. In reality, sounds are not only characterized as purely voiced or unvoiced but also as a blend of both. This simplification, on the other hand, is beneficial for providing fundamental knowledge of one of the ways the source signal in the source-filter model might be modeled. By attenuating certain frequencies and accentuating others in the excitation signal's spectrum, the filter molds the source signal (much as the vocal tract "shapes" the air). The output signal's spectrum changes as a result of this filtering, with a new spectral envelope and formant structure. As we've seen, the source-filter paradigm attempts to mimic the acoustic production of speech.

All-pole and log-spectrum filters are the most often utilized filter models. All-pole models are implemented with linear predictive coding (LPC), and log-spectrum filters are implemented with mel-log spectrum approximation (MLSA)(Imai et al., 1983). The figure below shows the representation of a source-filter model of speech production.

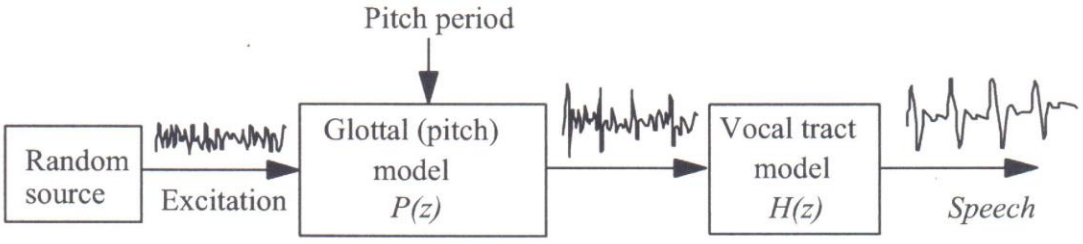


Figure 2. 2: A source-filter representation of speech production (Doshi, 2021).

2.1.2 SPEECH FEATURE EXTRACTION AND MAPPING

We construct vocoding parameters from speech analysis that typically include spectral and prosodic components to represent the input speech. The vocoding settings encode the speech in such a way that it may be reconstructed later on after transmission. The vocoding parameters are further turned into speech features, which we refer to as feature extraction, to improve the efficacy of voice conversion by modifying the acoustic qualities.

With regards to the spectral component, feature extraction seeks to derive low-dimensional representations from the high-dimensional raw spectra. In general, spectral properties are capable of accurately representing a speaker's uniqueness. Not only must the feature suit the spectral

envelope well, but it must also be reversible. They should exhibit good interpolation features that enable them to be easily modified. (Sisman et al., 2020)

The magnitude spectrum can be shifted to the Mel or Bark frequency scales for perceptually significant voice conversion. Additionally, it can be translated to the cepstral domain using a finite number of coefficients through the log-magnitude Discrete Cosine Transform. The correlation between the cepstral coefficients is lower. Thus, the magnitude spectrum of a high dimension is turned into a representation of a standard dimension feature. Mel-cepstral coefficients (MCEP), linear predictive cepstral coefficients (LPCC), and line spectral frequencies are some of the most frequently used speech features (LSF). Typically, We extract the features from these frames and end up with a feature vector for each frame. The mel frequency cepstral coefficients (MFCC) (Gupta et al., 2013) and the STRAIGHT spectrogram are two features that can be used (Zhang et al., 2009). These are the ones that are put to the test in this thesis.

In a typical voice conversion pipeline, feature mapping is used to modify speech features from the source to the target speaker. While spectral mapping aims to alter the timbre of the voice, prosody conversion aims to alter prosody characteristics such as fundamental frequency, intonation, and duration. Until now, spectral mapping has remained at the core of a large number of voice conversion experiments. (Sisman et al., 2020).

2.1.3 SPEECH RECONSTRUCTION

Speech reconstruction can be thought of as an inverse function of speech analysis, in which updated parameters are used to build an audible speech signal. It works in conjunction with speech analysis. After amplitude modification, a Griffin-Lim algorithm is utilized to reconstruct a speech signal from a modified short-time Fourier transform (Griffin and Lim, 1983). Because the speech reconstruction process affects the output speech quality, speech reconstruction is also a critical area of research in voice conversion.

2.2 VOICE CONVERSION

As discussed earlier in Chapter 1, Voice conversion could be regarded as a subfield within the broader area of voice transformation; It concerns transforming different aspects of a speech signal without changing its linguistic properties.

The entire voice conversion process in a typical VC system is often divided into two phases: training and conversion. The training phase involves conditioning the model on the speech features retrieved during the feature analysis stage and learning the feature mapping. During the conversion step, the trained model makes use of the knowledge gained during the training phase to adapt the source speakers' voices to sound like the target speakers' voices.

The VC systems are primarily classified into two categories: those that use Speech-to-Speech (STS) and those that use Text-to-Speech (TTS).

Additionally, VC models are classified into various types based on their approach to voice conversion. Parallel VC, non-parallel VC, VC based on mono-lingual speech data, VC based on cross-lingual speech data, intra-gender VC, cross-gender VC, one-to-one VC, and many-to-many VC are all examples.

Parallel and non-parallel VC systems are classified according to the linguistic content of the voice datasets used to train the VC modules. The training data for parallel VC systems consists of samples with identical linguistic contents (e.g., the same spoken words) from different speakers that are time-aligned (Mouchtaris et al., 2004). As a result of the frame-wise alignment of the linguistic contents, the speech feature mapping component can readily transfer the vocal features of source speakers to the vocal features of target speakers. In contrast, in non-parallel VC systems, training data consists of samples with misaligned source and target speaker linguistic contents.(Dhar, 2021) As a result of this misalignment, it becomes more difficult to map the vocal characteristics of the source speakers to the vocal characteristics of the target speakers than it is with parallel VC systems.

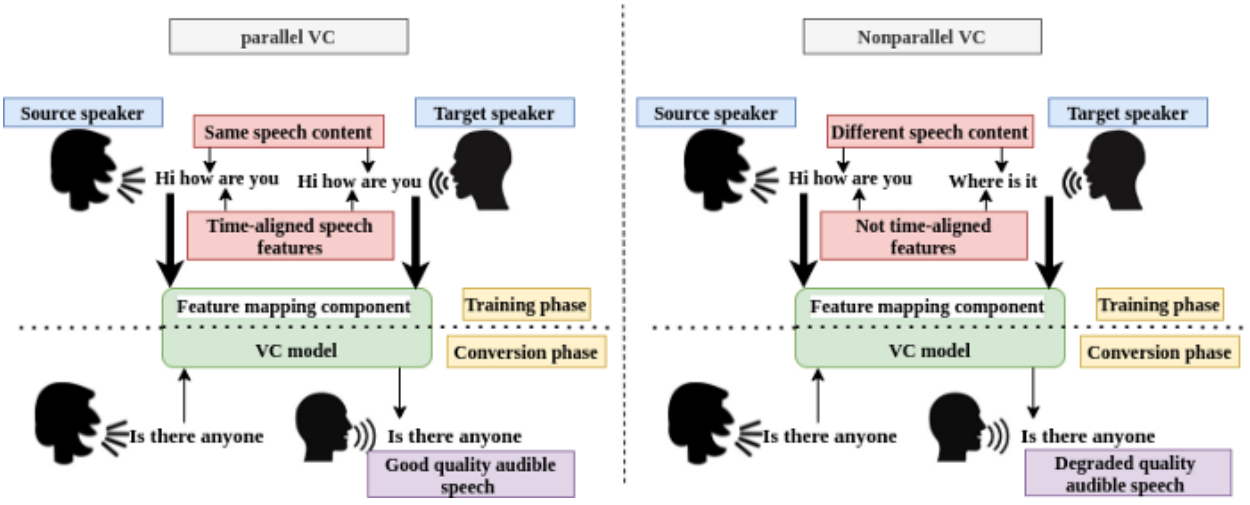


Figure 2. 3: Representation of the Parallel/Non-parallel Voice Conversion framework (Dhar, 2021)

Secondly, voice conversion techniques can be classified according to whether or not they require speech transcriptions. Thirdly, they can be classified according to their ability to convert across languages.

Additionally, voice conversion systems vary in their degree of reliance on the source and target speakers (s). As a result, they can be classified further into one-to-one systems, many-to-one systems, and many-to-many systems, among others, based on how they map sounds. A one-to-one system converts just between two speaker identities, whereas a many-to-one system can train on several sources to convert the speech signal to a single target voice (Mohammadi and Kain, 2017). The requirement for parallel data, transcripts, language consistency, and the appearance of the mapping between source and target speakers all have a significant impact on the system's usability. Additionally, because parallel- and transcribed data may be unavailable, the system's characteristics dictate the amount of effort required to acquire and preprocess data for the system.

VC systems are also classified as intra-gender VC systems or cross-gender VC systems, depending on the gender of the speakers involved in the voice conversion process. Male and female speakers have different Mel-scaled power spectrograms, Mel-frequency cepstral coefficients (MFCCs), power spectrogram chroma, spectral contrast, and tonal centroid properties. This implies that the speaker's gender has a significant part in the voice conversion process.(Dhar, 2021) Intra-gender VC occurs when both the source and target speakers are of the same gender, whereas cross-gender VC occurs when both the source and target speakers are of the opposite gender. Due to the closeness of gender-dependent speech patterns, intra-gender VC allows for a more straightforward mapping of speech features than cross-gender VC. Cross-gender VC models, on the other hand, are more robust because they may be utilized for both intra- and cross-gender voice conversion (Dhar, 2021)

The other category which is the TTS based VC method is categorized into two; ASR module-independent TTS-based VC systems and ASR module-dependent TTS-based VC systems. TTS-based VC systems based on the ASR module are combined with TTS synthesis models that have

been pre-trained. Using TTS systems allows speech to adhere to the linguistic content. In models that use the transfer learning technique, attention mechanisms are used to share the knowledge and decoder architecture of the TTS synthesis model with the speech-to-speech conversion based encoder-decoder model. In TTS-based VC systems that rely on ASR modules, an additional ASR module is used. The ASR module takes the content of the source speakers' speech and utilizes the TTS synthesis model to create target speakers' speech from the content of the source speakers' speech. (Dhar, 2021)

This thesis applies the method of a non-parallel many to many, cross-gender voice conversion (VC) using Generative adversarial networks (GAN) which is discussed further in the sections below.

2.3 ARTIFICIAL NEURAL NETWORKS (ANN)

In recent years, advancements in machine learning methodologies and hardware have enabled the use of artificial neural networks (ANNs) to handle complex problems very quickly (LeCun et al., 2015). Additionally, ANNs have been demonstrated to be effective in the setting of voice conversion. A pioneering example is (Narendranath et al., 1995), which employed an ANN to change the formant frequencies of many speakers and synthesized speech using a formant vocoder.

Recent attempts to convert speech to text using ANNs have utilized feed-forward topologies. A feedforward neural network is a straightforward acyclic architecture in which information goes from the input nodes to the hidden nodes and then to the output nodes. Each neuron has a distinct activation function, such as the sigmoid function, and the network as a whole is trained using backpropagation (Haykin et al., 2001). Desai et al. (Desai et al., 2010) first extracted features from the speech input and utilized them for training a four-layer feedforward network mapping the signal between speakers. Additionally, voice conversion using autoencoders has been proposed.

An autoencoder is a sort of neural network that has been trained to replicate its input by first encoding the data to a sparse representation and then reconstructing the original input from the encoded data. As a result, the model is compelled to learn the data's most beneficial properties. (Goodfellow et al., 2016) Mohammadi and Kain (Mohammadi and Kain, 2014) employed the

compact representation obtained from source data as mapping characteristics in applying this type of architecture for voice conversion. They then trained a deep feedforward architecture to map the speech signal between different speakers. Additionally, voice conversion has been proposed using recurrent neural networks. Recurrent neural networks are a subclass of ANNs that are distinguished by their ability to evaluate fresh input in the context of past input, implying that they have memory. They contain a loop in this sense, as old input is fed back into the network with fresh input, affecting the output in the process (Jain and Medsker, 1999). Sun et al. (Sun et al., 2015) used a recurrent architecture to model both the long- and short-term temporal dependencies between a source and target voice.

As seen in the preceding paragraphs, numerous ways to voice conversion with ANNs exist. However, because this work will focus on GANs for voice conversion, the concepts stated above will not be discussed.

2.4 DEEP LEARNING

As indicated previously in Section 2.3, approaches for voice conversion using artificial neural networks have been presented. Additionally, the rise of deep learning has resulted in a significant improvement in the performance of these methods. There has been recent works in non-parallel voice conversion based on deep neural network models which are broadly classified into TTS-based approaches which have been discussed above, auto-encoder-based approach as described in (Ding and Gutierrez-Osuna, 2019; Qian et al., 2020, 2019) aims to encode speaker-independent information from input audio by using correct constraints to train models. To remove speaker-dependent information, this method necessitates carefully established constraints, and the converted speech quality is determined by how much linguistic information can be extracted from the latent space, and GAN-based approaches (Li et al., 2021).

Additionally, previous research has demonstrated the viability of employing GANs for voice conversion. This work employs the method of GANs for non-parallel intragender voice conversion for the same language, i.e., monolingual; it is discussed in the following sections.

2.5 GENERATIVE ADVERSARIAL NETWORKS (GAN)

Ian J. Goodfellow and colleagues introduced Generative Adversarial Networks (GANs) in 2014 (Goodfellow et al., 2014). In machine learning, GANs are used to conduct unsupervised learning tasks. It is composed of two models that detect and learn patterns in input data automatically.

The two models are referred to as Generator and Discriminator, respectively. They compete to investigate, capture, and duplicate the variations included in a dataset. GANs can be used to generate new instances that could have been plausibly drawn from the original dataset.

In GANs, a Generator is a neural network that generates fictitious data for training on the discriminator. It acquires the ability to fabricate credible data. The generated examples/instances are used to train the discriminator on negative examples. It generates a sample from a fixed-length random vector that contains noise. The Generator's primary objective is to convince the discriminator that its output is genuine. By assessing the weight's impact on the output, the backpropagation method is utilized to modify each weight in the proper direction. It is also used to obtain gradients, which can be used to modify the generator weights.

The Discriminator is a neural network that distinguishes genuine data from the Generator's bogus data. When the discriminator is trained, it is connected to two loss functions. The discriminator disregards the generator loss during discriminator training and focuses exclusively on the discriminator loss.

While training the discriminator, it classifies both real and generated data. The discriminator loss penalizes the discriminator for misclassifying either actual or fraudulent data instances as real.

The discriminator network's weights are updated by backpropagation from the discriminator loss.

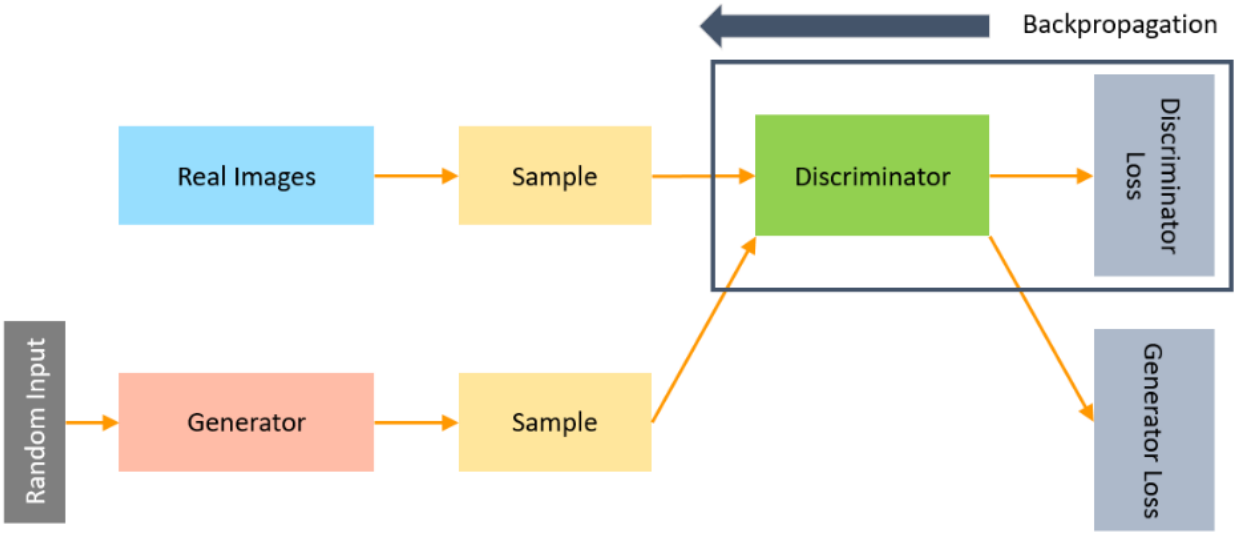


Figure 2. 4: Typical architecture of a Generative Adversarial Network, illustrating the link between the generator and the discriminator (Goodfellow et al., 2014).

While GANs are effective for picture production, they have also been used successfully for a variety of speech processing applications in recent years (Kaneko et al., 2017a, 2017b). A non-parallel VC technique based on a GAN variant termed cycle-consistent GAN (Cycle-GAN) was recently published (Kaneko and Kameoka, 2017), which was originally presented as a method for translating images utilizing unpaired training examples (Yi et al., 2017; Zhu et al., 2017).

This method, which we refer to as CycleGAN-VC, is designed to learn the mapping G of acoustic features between one attribute X and another Y, as well as its inverse mapping F and a discriminator D for distinguishing the acoustic features of converted speech from those of real speech, via a training loss that combines an adversarial and a cycle consistency loss. While this method has been demonstrated to perform pretty well, it has a significant restriction in that it only learns one-to-one mappings. It is desirable to obtain many-to-many mappings in a variety of VC application settings.

Another approach, termed StarGAN, that enables non-parallel many-to-many voice conversion (VC), was introduced lately (Kameoka et al., 2018), which was an extension of CycleGAN-VC. It utilizes a variant of a generative adversarial network (GAN) to accomplish this. This method, which we refer to as StarGAN-VC, is notable in that it (1) does not require parallel utterances, transcriptions, or time alignment procedures for speech generator training, (2) simultaneously

learns many-to-many mappings across multiple attribute domains using a single generator network, (3) generates converted speech signals quickly enough for real-time implementations, and (4) requires only a few minutes of training examples to generate reasonably converted speech signals (Kameoka et al., 2018). This method produced higher sound quality and speaker similarity in the speaker identity conversion task which brings us to the method used in this thesis.

This thesis uses an unsupervised non-parallel many-to-many voice conversion (VC) method called StarGANv2 which is in relation to the work by Li et al (Li et al., 2021)using GANs, which was just recently introduced and is discussed further in the section below:

2.6 STARGANv2-VC

StarGANv2 is a subtype of generative adversarial networks that has been mostly utilized for style transfer from image to image. It provides an unsupervised non-parallel many-to-many voice conversion (VC) approach based on a generative adversarial network (GAN) in this paper (Li et al., 2021). By combining adversarial source classifier loss and perceptual loss, this model beats earlier VC models significantly. It is generalizable to a variety of voice conversion problems, including any-to-many conversion, cross-lingual conversion, and singing conversion. Additionally, this framework may transform basic reading speech to stylistic speech, such as emotional and falsetto speech, via a style encoder. This framework generates natural-sounding speech and exceeds the prior state-of-the-art method, AUTO-VC (Qian et al., 2019) and StarGAN-VC (Kameoka et al., 2018), in terms of both naturalness and speaker similarity.

StarGANv2 enables the conversion of plain speech to speech with a variety of styles, introduces a novel adversarial source classifier loss that significantly improves the similarity of converted and target speech in terms of speaker identity, and is the first voice conversion framework that we are aware of that employs perceptual losses via both automatic speech recognition (ASR) and fundamental frequency (F0) extraction networks.

StarGANv2 employs a single discriminator and generator to generate varied images in each domain using either the style encoder or the mapping network's domain-specific style vectors. To accomplish F0-consistent conversion, we applied the same architecture to voice conversion,

regarded each speaker as a separate domain, and incorporated a pre-trained joint detection and classification (JDC) F0 extraction network (Kum and Nam, 2019).

A summary of this framework is presented in greater detail below.

- **Generator:** The generator G transforms an input mel-spectrogram into G (X_{sr}, h_{sty}, h_{f0}), which reflects the style in h_{sty}, which is provided by either the mapping network or the style encoder, and the fundamental frequency in h_{f0}, which is provided by the convolution layers in the F0 extraction network F (Li et al., 2021).
- **F0 network:** The network used to extract F0, F is a to extract the purpose of extracting the fundamental frequency from an input mel-spectrogram. Convolutional layers precede the BLSTM units in the JDC network. As input features, we use solely the convolutional output $styled$for X $\in \mathcal{X}$ (Li et al., 2021)
- **Mapping network:** The mapping network M generates a style vector $h_M = M(z, y)$ in a domain $y \in \mathcal{Y}$ with a random latent code $z \in \mathcal{Z}$. The latent coding is drawn from a Gaussian distribution to generate a variety of style representations across domains. The style vector representation is shared across all domains until the final layer, at which point the common representation is subjected to a domain-specific projection (Li et al., 2021).
- **Discriminators:** In the paper (Choi et al., 2020), the discriminator D has shared layers that learn the common properties between genuine and fake samples across domains, followed by a domain-specific binary classifier that classifies if a sample is real in each domain $y \in \mathcal{Y}$. However, because the domain-specific classifier only has one neural layer, it may miss out on critical domain-specific features like a speaker's pronunciations. To solve this issue, we construct an additional classifier C that learns the original domain of transformed samples and has the same architecture as D. The classifier can provide feedback regarding attributes invariant to the generator but distinctive of the original domain, which the generator should enhance to generate a more similar sample in the target domain (Li et al., 2021), by learning what features evade the input domain even after conversion.

The figure below represents the StarGANv2 framework as discussed above;

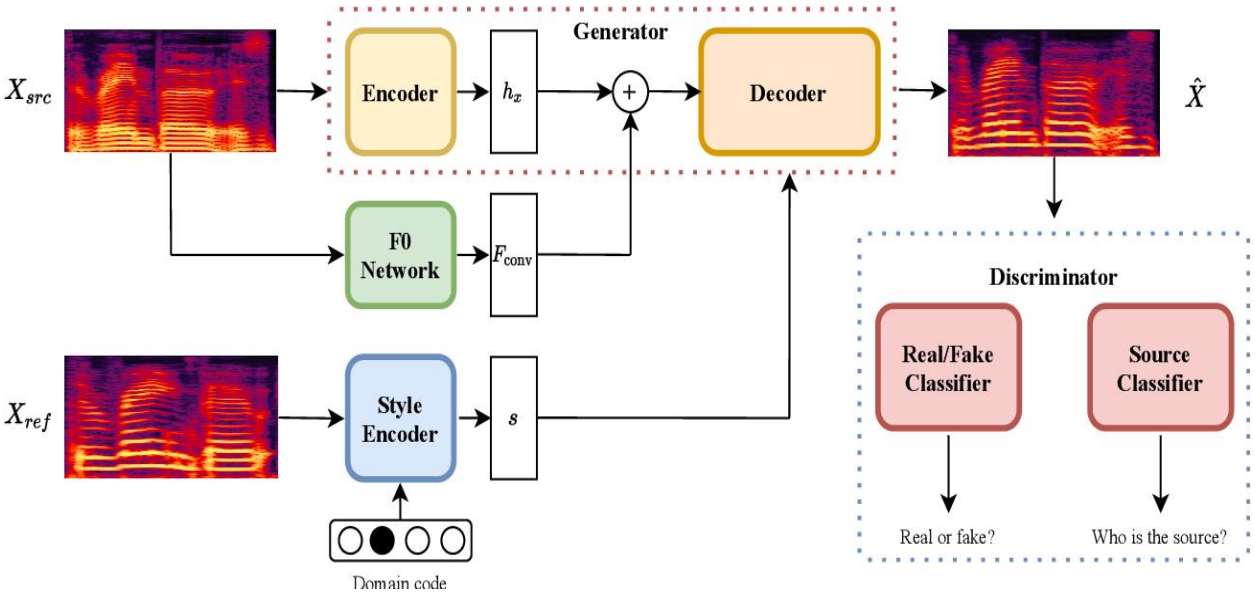


Figure 2. 5: StarGANv2-VC with style encoder. X_{src} is the source input, X_{ref} is the reference input that contains the style information, and $\widehat{X}$ represents the converted mel-spectrogram (Li et al., 2021)

2.7 STARGANV2-VC TRAINING

The purpose of StarGANv2-VC is to discover a mapping $G: \mathcal{X}_{y_{src}} \rightarrow \mathcal{X}_{y_{trg}}$ that transforms a sample $\boldsymbol{X} \in \mathcal{X}_{y_{src}}$ from the source domain $y_{src} \in \mathcal{Y}$ to a sample $\widehat{\boldsymbol{X}} \in \mathcal{X}_{y_{trg}}$ in the target domain $y_{trg} \in \mathcal{Y}$ without using parallel data (Li et al., 2021)

As related to the work by Li et al(Li et al., 2021), We randomly choose a target domain $y_{trg} \in \mathcal{Y}$ and a style code $s \in S_{y_{trg}}$ during training using either a mapping network with $s = M(z, \mathcal{Y}_{trg})$ and a latent code $z \in \mathcal{Z}$, or a style encoder with $s = S(\boldsymbol{X_{ref}}, \mathcal{Y}_{trg})$ and a reference input. We train our model using the following loss functions given a mel-spectrogram $\boldsymbol{X} \in \mathcal{X}_{y_{src}}$, a source domain $y_{src} \in \mathcal{Y}$, and a target domain $y_{trg} \in \mathcal{Y}$.

- **Adversarial Loss:** The generator takes an input melspectrogram **X** and a style vector s and uses the adversarial loss to learn how to generate a new melspectrogram G(**X,** s).

$$\mathcal{L}_{adv} = \mathbb{E}_{\boldsymbol{x},y_{\boldsymbol{src}}}[logD(\boldsymbol{X}, y_{\boldsymbol{src}})] + \mathbb{E}_{\boldsymbol{x},y_{\boldsymbol{trg}},s}\left[\log\left(1 - D\big(G(\boldsymbol{X}, s), y_{trg}\big)\right)\right] \quad (1)$$

Where $D(.\ , y) =$ output of real/fake classifier for the domain $y \in \mathcal{Y}$.

- **Adversarial source classifier loss:** With the source classifier C, we combine an adversarial loss function.

$$\mathcal{L}_{advcls} = \mathbb{E}_{\boldsymbol{x},y_{\boldsymbol{trg}},s}\left[CE(C\left(G(\boldsymbol{X},s),y_{trg}\right))\right] \quad (2)$$

Where $CE(.)$ =cross-entropy loss function.

- **Style reconstruction loss:** The style reconstruction loss is used to ensure that the generated samples may be used to rebuild the style code.

$$\mathcal{L}_{sty} = \mathbb{E}_{\boldsymbol{x},y_{\boldsymbol{trg}},s}\left[\| \ s - S\left(G(\boldsymbol{X},s),y_{trg}\right) \|_1\right] \quad (3)$$

- **Style diversification loss:** To force the generator to generate multiple samples with varied style codes, the style diversification loss is maximized. We maximize MAE of the F0 features between samples created with different style codes in addition to the mean absolute error (MAE) between generated samples.(Li et al., 2021)

$$\mathcal{L}_{ds} = \mathbb{E}_{\boldsymbol{x},s_1s_2,y_{\boldsymbol{trg}}}\left[\| \ \left(G(\boldsymbol{X},s_{\boldsymbol{1}}) - G(\boldsymbol{X},s_{\boldsymbol{2}})\right) \|_1\right] + \mathbb{E}_{\boldsymbol{x},s_1s_2,y_{\boldsymbol{trg}}}[\| \ F_{conv}(G(\boldsymbol{X},s_{\boldsymbol{1}})) - F_{\boldsymbol{conv}}(G(\boldsymbol{X},s_{\boldsymbol{2}}))) \|_1] \quad (4)$$

Where $s_1, s_2 \in S_{y_{trg}}$ = two randomly sampled style codes from domain $y_{trg} \in \mathcal{Y}$.

$F_{conv}(.)$ = output of convolutional layers of F0 network F.

- **F0 consistency loss:** To obtain F0-consistent findings, we combine an F0-consistent loss with the F0 network's normalized F0 curve. F(X) returns the absolute F0 value in Hertz for each frame of an input mel-spectrogram X. (Li et al., 2021)Due to the fact that the average F0 of male and female speakers is different, we normalize the absolute F0 values F(X) by their temporal mean, denoted by: $\hat{F}(\boldsymbol{X}) = \frac{F(\boldsymbol{X})}{\|F(\boldsymbol{X})\|_{\boldsymbol{1}}}$ (Li et al., 2021)**.** The F0 consistent loss is given as:

$$\mathcal{L}_{f0} = \mathbb{E}_{\boldsymbol{x},s}\left[\| \ \hat{F}(\boldsymbol{X}) - \hat{F}\left(G(\boldsymbol{X},s)\right) \|_1\right] \quad (5)$$

- **Speech consistency loss:** To ensure that the converted speech has the same linguistic content as the source, we utilize a speech consistency loss based on convolutional features extracted from a pretrained joint CTC-attention VGG-BLSTM network(Kim et al., 2017) available in Espnet toolbox (Watanabe et al., 2018). As in (Polyak et al., 2020), we use the output of the intermediate layer preceding the LSTM layers as the linguistic feature signified by $h_{asr}(.)$. The term "speech consistency loss" refers to

$$\mathcal{L}_{asr} = \mathbb{E}_{\boldsymbol{x},s}\left[\| \ h_{asr}(\boldsymbol{X}) - h_{asr}\left(G(\boldsymbol{X},s)\right) \|_1\right] \text{ (Li et al., 2021)} \quad (6)$$

- **Norm consistency loss:** We apply the norm consistency loss to retain the generated samples' speech/silence intervals. The absolute column-sum norm is defined as follows for a mel-spectrogram X with N mels and T frames at the t^{th} frame**:** $\| \boldsymbol{X}., t \| = \sum_{n=1}^{N} |\boldsymbol{X}_{n,t}|$, where $t \in \{1, \dots, T\}$ =frame index. (Li et al., 2021)The norm consistency loss is then given by: $\mathcal{L}_{norm} = \mathbb{E}_{\boldsymbol{x},s}\left[\frac{1}{T}\sum_{t=1}^{T}|\| \boldsymbol{X}., t \| - \| G(\boldsymbol{X}, s))., t \||\right]$ (7)
- **Cycle consistency loss:** Finally, we applied the cycle consistency loss to preserve all other features of the input.

  $$\mathcal{L}_{cyc} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{y_{src}},y_{trg},s}[\| \boldsymbol{X} - G(G(\boldsymbol{X}, s), \tilde{s}) \|_1] \quad (8)$$

  Where $\tilde{s} = S(\boldsymbol{X}, y_{src}) =$ the estimated style code of the input in the source domain $y_{src} \in \mathcal{Y}$.
- **Summary of objectives:** The following summarizes our whole generator objective functions.

  $$\min_{G,S,M} \mathcal{L}_{adv} + \lambda_{advcls}\mathcal{L}_{advcls} + \lambda_{sty}\mathcal{L}_{sty} - \lambda_{ds}\mathcal{L}_{ds} + \lambda_{f0}\mathcal{L}_{f0} + \lambda_{asr}\mathcal{L}_{asr} + \lambda_{norm}\mathcal{L}_{norm} + \lambda_{cyc}\mathcal{L}_{cyc} \quad (9)$$

  Where $\lambda_{advcls}, \lambda_{sty}, \lambda_{ds}, \lambda_{f0}, \lambda_{asr}, \lambda_{norm}\ and\ \lambda_{cyc} =$ hyperparameters for each term. The full objective of the discriminators is given as:

  $$\min_{C,D} -\mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}_{cls} \quad (10)$$

  Where $\lambda_{cls} =$ hyperparameter for source classifier loss, $\mathcal{L}_{cls}$ which is given as:

  $$\mathcal{L}_{cls} = \mathbb{E}_{\boldsymbol{x},y_{\boldsymbol{src}},s}[CE(C(G(\boldsymbol{X}, s), y_{src}))] \quad (11)$$

2.8 SPECTOGRAMS AND CEPSTRAL COEFFICIENTS

Mel-Frequency Cepstral Coefficients are the intermediate form that the model explored in this research utilizes to represent the spoken signal (MFCCs). This is a notion linked to spectrograms. A spectrogram is a representation of an audio signal in the frequency domain created by performing a Fourier transform on overlapping windowed time segments (Smith, 2007). Additionally, the amplitude is converted to decibels and the frequency axis to log scale and then transferred to the Mel scale to create a Mel-spectrogram. The Mel scale is used to establish a more precise link between the spectrogram and human perception of sound. Additionally, the Mel scale emphasizes lower frequencies since they communicate more information about speech than higher frequencies, which are dominated by noise.

The form of a human's vocal tract, which comprises the lips, throat, and tongue, for example, filters the sounds created by the human. This form is manifested in the power spectrum's spectral envelope, and by precisely representing the envelope, we may deduce the phoneme produced by the human. The MFCCs can be obtained as the amplitudes of the resulting cepstrum by treating the log power spectrum on a Mel frequency scale (for a windowed time segment) as a signal and applying the discrete cosine transform to it (Koolagudi et al., 2012). These are advantageous for speech analysis in a variety of ways since they contain information on the formants, phonemes, spectral envelope, and other aspects of the speech signal. They are frequently used to parametrize speech and as machine learning training features (On et al., 2006).

2.9 VOCODERS

A vocoder can be used to generate an audio signal from a Mel spectrogram or a collection of cepstral coefficients. Griffin-Lim and the WaveNet neural vocoder are both algorithms capable of performing this task. To evaluate the results of the voice conversion subjectively, one may wish to listen to synthetic speech generated from the converted cepstral coefficients.

To accomplish this, the transformed speech will be synthesized using the Parallel WaveGan (Yamamoto et al., 2020) Vocoder for this work which is a distillation-free, fast, and small-footprint waveform generation method using a generative adversarial network.

CHAPTER 3

METHODOLOGY

This chapter discusses how the theory presented in Chapter 2 can be implemented, as well as the many system versions that were tested. The work presented in this thesis was accomplished through the use of a variety of frameworks and technologies.

The many tools that were employed and the rationale behind their selection will be discussed below. As noted in Chapter 2, VC entails multiple distinct processes, both in terms of training and conversion. Different technologies were utilized for various components.

In order to carry out this voice conversion task, different tools and deep learning networks were used as well as the dataset for the ASR and F0 models and the Speech Corpus used which would be discussed in the sections below.

3.1 TOOLS USED

This work was mainly based on machine learning architecture which was performed using various deep learning libraries via the Ubuntu Debian Linux Distribution software, which is an open-source and free software with better computing speed and greater memory and can be used both remotely and in desktop environments. The Ubuntu software can be used with several environments and operating systems, but for this work, We used the Jupyter environment on Anaconda and used Python as the programming language due to its simplicity. Some of the deep learning libraries attached to this work are as follows:

- **PyTorch:** PyTorch is a highly optimized tensor library designed for GPUs and CPUs in Deep Learning applications. It is a Python-based open-source machine learning package developed mainly by the Facebook AI Research team. It is a popular machine learning library, alongside TensorFlow and Keras. PyTorch is a Python package that uses the torch library to perform tensor computations on Graphics Processing Units. Currently, it is the most widely used library in the field of deep learning and artificial intelligence research.
- **Librosa:** Librosa is a Python program that facilitates music and audio analysis. It enables the development of music information retrieval systems by providing the required

building elements. Since this work deals with voice samples, It was important to use librosa to analyse and also convert sound samples from different versions for example, .flac to .wav files.

- **Numpy:** Numpy is a Python library for scientific computing. It performs effectively with high-dimensional data, which necessitates the usage of Keras.
- **Kaldi:** Kaldi is a freely available, open-source toolset for conducting research on voice recognition. Kaldi provides a speech recognition system based on finite-state transducers (using the open-source OpenFst), as well as full documentation and scripts for constructing complete recognition systems. Kaldi is built in C++, and its core library allows modeling of any phonetic context sizes, as well as acoustic modeling with subspace Gaussian mixture models (SGMM) and regular Gaussian mixture models, as well as all commonly used linear and affine transforms. (Povey et al., 2011) .

3.2 SPEECH CORPUS

The Speech Corpus used for the experiments in this work was the CSTR VCTK Corpus (i.e., English Multi-Speaker Corpus for CSTR (Centre for Speech Technology) Voice Cloning Toolkit) contains voice data from 110 English speakers with a range of accents. Each speaker reads around 400 sentences from a newspaper, the rainbow passage ("The Rainbow Passage | IDEA," 2011), and the elicitation paragraph for the speech accent archive. With permission from Herald & Times Group, the newspaper texts were taken from Herald Glasgow. Each speaker is presented with a unique selection of newspaper articles chosen using a greedy algorithm that maximizes contextual and phonetic coverage. The paper describes the details of the text selection algorithms by, all speakers are given the identical rainbow passage and elicitation text. International Dialects of English Archive, has the rainbow passage. The elicitation paragraph is identical to that of the speech accent archive (http://accent.gmu.edu). All speech data was captured in a semi-anechoic environment at the University of Edinburgh using a comparable recording setup: an omnidirectional microphone (DPA 4035) and a small diaphragm condenser microphone with a wide dynamic range (Sennheiser MKH 800). (However, two speakers, p280 and p315, experienced technical difficulties with the MKH 800 audio recordings. Each recording was converted to 16 bits, downsampled to 48 kHz, then end-pointed manually. This corpus was originally intended for use in HMM-based text-to-speech synthesis systems, more specifically

for speaker-adaptive HMM-based speech synthesis systems that employ average voice models trained on numerous speakers and speaker adaption technologies. Additionally, this corpus is well-suited for neural network-based multi-speaker text-to-speech systems and neural waveform modeling.

3.3 DATA PREPROCESSING

In order to carry out the voice conversion process, The data and audio files have to be preprocessed before being fed into the StarGAN architecture. The processes carried out were as follows:

1. **Loading the Audio files**: The Input data which is used as the baseline model, consists of audio files of spoken speech in .wav format. The files were read and loaded into a Numpy 2D array which consisted of a sequence of numbers, each representing a measurement of the intensity or amplitude of the sound at a particular moment in time (Doshi, 2021). The number of such measurements is determined by the sampling rate, which is 48KHz.
2. **Conversion to the uniform sample rate, channels, and duration**: Because our deep learning models assume that all of our input items are the same size, we now execute some data cleaning operations to standardize the dimensions of our audio data (Doshi, 2021). We resample the audio to 24KHz to ensure that each component is sampled at the same rate. All objects are converted to the same number of channels. Additionally, all elements must be transformed to the same audio duration. This is accomplished by padding shorter sequences and truncating larger sequences.
3. **Data Augmentation of raw audio:** Additionally, we used data augmentation approaches to increase the variety of our input data and assist the model in generalizing to a broader range of inputs. This also included randomly Time Shifting our audio left or right by a small percentage, as well as adjusting the pitch or speed of the audio by a little amount.
4. **Conversion to Mel Spectrograms:** This raw audio is now converted to Mel Spectrograms which captures the nature of the audio as an image by decomposing it into the set of frequencies that are included in it.
5. **Conversion to MFCC:** The Mel Spectrograms were converted to MFCCs since it produces a compressed representation of the Mel Spectrogram by extracting only the most

essential frequency coefficients, which correspond to the frequency ranges at which humans speak.

Following the conversion of the original raw audio file to Mel Spectrogram (or MFCC) pictures, data cleaning and augmentation are performed.

This provides us with the features of our input and the labels for our target. This data is now prepared for incorporation into our deep learning model.

3.4 TRAINING OF THE ASR MODEL

The deep learning architecture use for training the ASR (Automatic Speech Recognition) consisted of a Convolutional Neural Network (CNN) plus RNN-based (Recurrent Neural Network) architecture that uses the CTC loss algorithm to separate each character of the words in the speech. The model consisted of a few blocks:

- A convolutional network composed of a few Residual CNN layers that process the input spectrogram images and generate feature maps from them (Doshi, 2021)

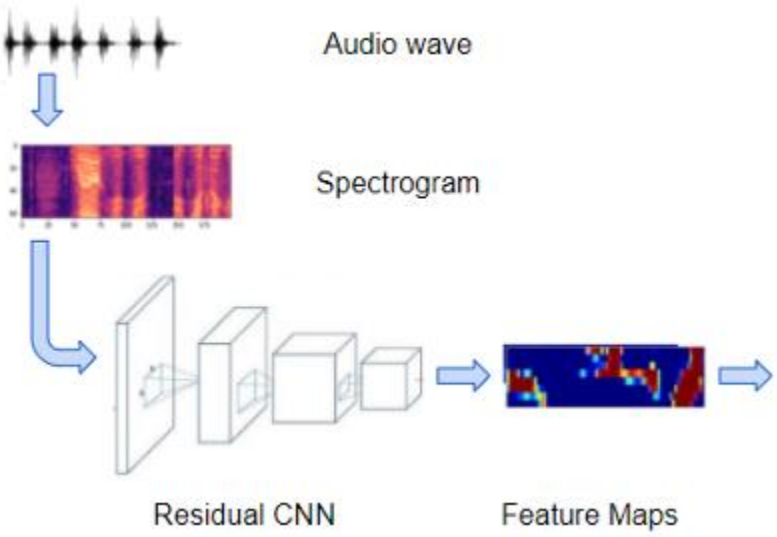


Figure 3. 1: Mapping of the Spectrogram images using CNN Network (Doshi, 2021)

- A recurrent network composed of several bidirectional LSTM layers that process the feature maps in a succession of separate timesteps or 'frames' that correspond to the desired sequence of output characters. (An LSTM is a highly frequent sort of recurrent

layer; the abbreviation stands for Long Short Term Memory.) In other words, it translates the continuous feature maps that represent the audio to a discrete representation.

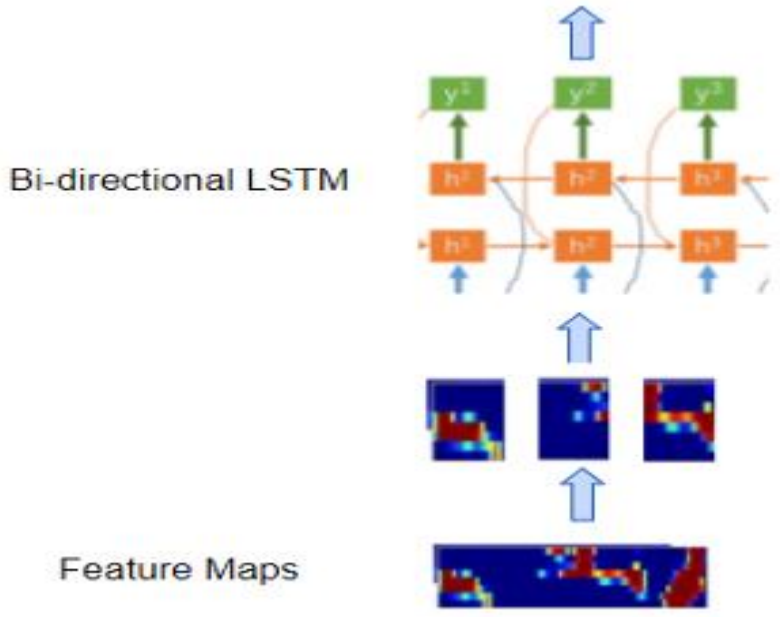


Figure 3. 2: Processing frames from the feature maps using the RNN network (Doshi, 2021)

- A softmax linear layer that utilizes the LSTM outputs to generate character probabilities for each timestep of the output.
- Additionally, linear layers exist between convolutional and recurrent networks and aid in reshaping the outputs of one network into the inputs of the other.

Our objective is to map those timesteps, or 'frames,' to specific characters in our target transcript, but we are unsure of the appropriate number of frames, the location of each frame's boundaries, or how to match the audio with each letter in the text transcript as the audio and spectrogram images are not segmented in advance to provide us with this data. This creates an issue and makes good ASR training more difficult. To address the issue, we used the CTC Algorithm to train it further and align the sequences following the previous phases. This is covered in further detail further down.

3.5 CTC ALGORITHM

CTC is used to align the input and output sequences when the input is continuous, and the output is discrete, and there are no obvious element boundaries to map the input to the output sequence elements (Doshi, 2021).

What makes this unique is that it achieves this alignment automatically, rather than needing you to manually include it in the labeled training data.

As described previously, the convolutional network's output feature maps are divided into distinct frames and fed into the recurrent network. Each frame represents a different timestep in the original audio wave. However, when you create the model, you choose the number of frames and the time of each frame as hyperparameters. The recurrent network, followed by the linear classifier, then predicts probability values for each character in the vocabulary for each frame (Doshi, 2021).

The CTC algorithm's task is to take these character probabilities and derive the correct character sequence.

To assist it in dealing with alignment issues and repeated characters, it incorporates the concept of a 'blank' pseudo-character (denoted by "-") into the vocabulary. As a result, the network's output of character probabilities includes the possibility of a blank character in each frame (Doshi, 2021).

The CTC algorithm works in two (2) modes:

- **CTC Loss (During Training):** It starts with a ground truth target transcript and attempts to train the network in such a way that the likelihood of producing that right transcript is maximized. The Loss is calculated as the likelihood of the network correctly anticipating the sequence. To accomplish this, the algorithm generates a list of all potential sequences predicted by the network and then chooses the subset that corresponds to the target transcript.
  A detailed explanation of its sequences is illustrated in the figure below:

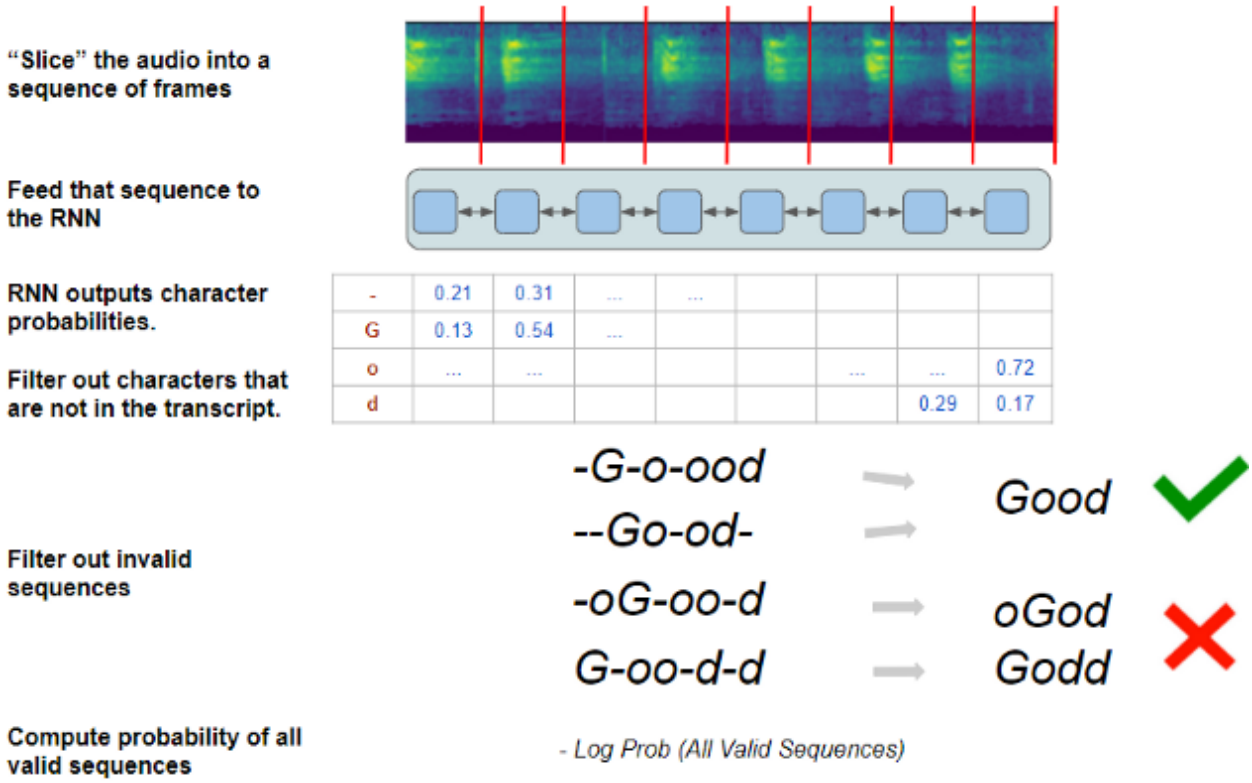


Figure 3. 3: CTC Loss Algorithm (Doshi, 2021)

With these limitations in place, the algorithm now has a set of valid character sequences that result in the right target transcript.

It then calculates the overall likelihood of generating all of those valid sequences using the individual character probabilities for each frame. The network's purpose is to discover how to optimize that probability and so minimize the likelihood of generating any erroneous sequence (Doshi, 2021)

Because a neural network is designed to minimize loss, the CTC Loss is defined strictly as the negative log probability of all valid sequences (Doshi, 2021). As the network minimizes this loss during training, it adjusts all of its weights to generate the proper sequence.

- **CTC Decoding (During Inference):** Here, we lack a reference transcript and must guess the most likely sequence of characters. A detailed explanation of this process is shown in the figure below:

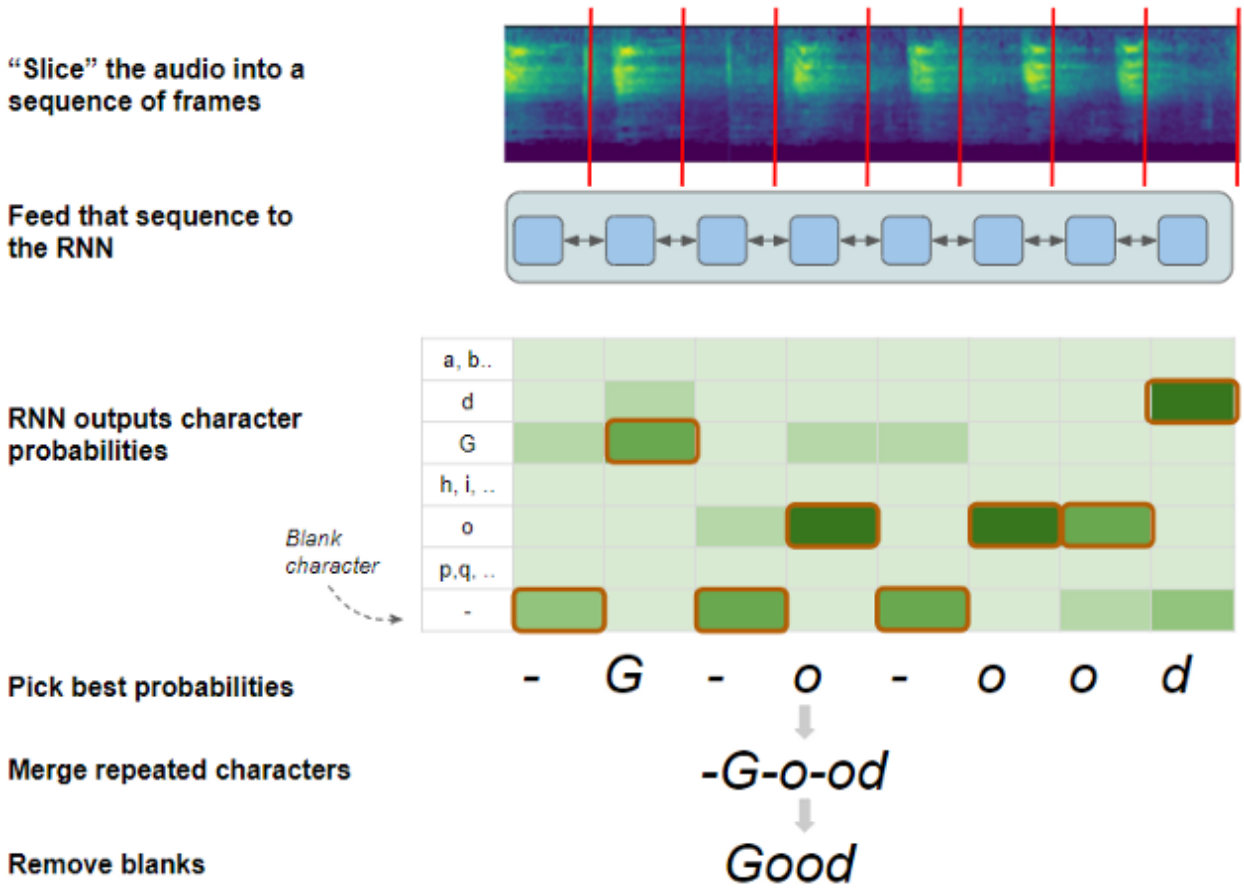


Figure 3. 4: CTC Decoding Algorithm (Doshi, 2021).

3.6 WORD ERROR RATE (WER)

Word Error Rate (WER) (Wikipedia, 2020) is a metric used to determine the degree of similarity between a reference text and a prediction. It is calculated as the ratio of each pair's numerator and denominator.

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

Where;

- S = number of substitutions;
- D = number of deletions;
- I = number of insertions;
- C = number of correct words;
- N = number of words in the reference (N = S+D+C)

CHAPTER FOUR

EXPERIMENTS AND RESULTS

This chapter discusses how the experiments, hardware, tools, data preprocessing, and hyperparameters were carried out. Finally, we discuss the result obtained from running the experiments.

4.1 EXPERIMENTAL PLAN

These experiments involve the training and testing using the StarGANv2 framework on randomly selected 20 speakers from the VCTK dataset, which comprises English speakers with various accents. The framework was trained on two randomly chosen speakers in four (4) different ways, as shown in the table with different accents and emotions. The ASR and F0 (which is from a JDC network) for this work were trained using the TIMIT dataset (Garofolo, John S. et al., 1993) which is made up of recordings of 630 speakers of eight (8) dialects of American English. The dataset was resampled to 24kHz and split randomly according to an 80%/10%/10% of train/Val/test partition. The data used for the experiments and how they were represented is shown in the table below:

GROUP	ID	GENDER	ACCENT
DATA 1	P232	M	English
	P248	F	Indian
DATA 2	P228	F	English
	P263	M	Scottish
DATA 3	P279	M	English
	P329	F	American
DATA 4	P376	M	Indian
	P295	F	Irish

Table 4. 1: Speaker Information of Data used for experiments.

4.2 EXPERIMENTAL SETUP

This section presents the setup used to conduct experiments. This includes the hardware, software, data prepossessing techniques used for model training and evaluation to ensure that our work is reproducible.

4.2.1 HARDWARE, SOFTWARE, AND ENVIRONMENT

Due to the computational requirements needed to perform subsequent development and testing of different models viably, I used a dedicated computer with the following specifications:

- Windows 10 PC
- Intel® Core™ i5-8250U CPU @ 1.60GHz 1.80GHz
- 64-bit Operating System, x64-based processor
- Installed memory of 8GB RAM
- TensorFlow GPU 1.13
- NVIDIA Driver
- CUDA Version 10.2
- Ubuntu 18.04 Unix Software

4.2.2 TRAINING DETAILS FOR THE EXPERIMENT

The StarGANv2 model trained our model for 150 epochs, a batch size of 5 with a save frequency of 2. For the preprocessing parameters, we set the sample rate to 24kHz. The spectrogram parameters were also under the preprocessing parameters: window length- 1200, hop length of 300, and number of FFT (fast Fourier transforms) to be 2048. I set the loss parameters as $\lambda_{sty} = 1$, $\lambda_{cyc} = 5, \lambda_{ds} = 1$, $\lambda_{norm} = 1$, $\lambda_{asr} = 10$, $\lambda_{f0} = 5$, $\lambda_{advcls} = 0.5$, which were all classified for the generator losses while the loss at the discriminator was given as $\lambda_{cls} = 0.1$. The source classifier joins the training process after 50 epochs. The optimizer used for this training configuration was the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate (LR) of 0.0001. All these configurations for the training are saved to a YAML file used as input during the conversion processes. The F0 model was trained with pitch contours given by World Vocoder (Morise et al., 2016) for 100 epochs, while the ASR model was trained at phoneme level for 80 epochs with a CER of 16.7%. The conversion was also carried out using the StarGAN-VC framework for a fair comparison.

4.2.3 IMPLEMENTATION AND CONVERSION DETAILS USING THE STARGANv2 FRAMEWORK

Before the conversion process, we first had to prepare the data from the VCTK Corpus for each of the data to generate a directory for the speaker set for each procedure, i.e., Data1, Data2,

Data3, and Data4. I imported the audio files in .wav format for each speaker in a set and generated the validation and train list text files to be also used for the training configuration. The process of the conversion took the following steps

- I then loaded the python packages used for this conversion, including torchaudio, librosa, yaml, munch, and the models from the ASR and F0, respectively.
- The speakers were then defined, which included two (2) speakers for each dataset that was to be used for the conversion, one male and female. We created a separate file for the data experiment.
- The speakers were converted to the melspectrogram, which is used as the source for the conversion
- The process for the mels and wave tensor was defined as the model to be built that included the mapping network and style encoder.
- We defined the speaker dictionary and also loaded the audio by using librosa at a sample rate of 24KHz
- The models were after that loaded, which included the F0 model from the JDC network, Vocoder using a pretrained parallel WaveGAN model, and the starganv2 model that involved a pretrained model and the configuration file stored in YAML, which was trained earlier while preparing the data.
- The input wave was then set by selecting the speaker as a reference and the particular audio and .wav path. In this case, we used the 23rd recording/audio of the speaker to be used as input.
- Finally, we then converted using the mapping network and the style network.

4.2.4 IMPLEMENTATION AND COVERSION DETAILS USING THE STARGAN FRAMEWORK

This framework was used to compare the two networks and hence was tested on just Data1 to determine which framework performs better. Unlike the StarGAN network, this process didn't involve the F0 network from JDC or the CTC ASR network. The procedure involved the following:

- Preprocessing of the Data using the Mel Cepstral Coefficients (MCEP) at a sample rate of 16kHz

- Training of the model where the batch size of 32, generator loss and discriminator loss of 0.0001, source and target speakers to be converted and no. of iterations of 200,000 was defined and loading of the audio/ wave path with librosa
- Finally, the conversion process and no. of .wav number for each speaker are specified. Also, no speakers for the speaker dictionary from where the two(2) speakers to be used for conversion are selected. The no. of speakers was set to 10 in this case, and the wav-path for each speaker was the 8th recording.

4.3 EVALUATIONS

The experiment for the StarGANv2 procedure took about 10 mins to give results and seconds for the conversion itself, while that of the StarGAN network took about 2hours to complete the conversion procedure.

In this work, objective and subjective evaluation metrics were utilized to assess the quality and similarity of the converted speech.

The objective criteria are determined automatically and are based on calculations on features derived from the evaluated speech. They benefit from repeatability, i.e., they always generate the same result given the same data, and they are inexpensive to run on a computer. However, as explained below, they are frequently unable to adequately replicate the perspective of a human listening to the same speech, and there are no valid objective metrics available for specific questions, such as grading the speech quality.

On the other hand, subjective criteria are based on the opinions of human listeners (subjects), making them more realistic as people use voice converter technology.

4.3.1 SUBJECTIVE METRICS

I randomly selected two speakers, one used as source and the other target speakers, which happens to be one male and one female, all from the 20 speakers trained. Both source and ground truth samples were chosen to have at least 5-second long audios so that there is enough information to judge the similarity and naturalness. I asked ten friends of mine, two from the same department, four from the music department, and four random people to judge /rate each audio clip on a scale of 1 to 5, where one indicates completely distorted and unnatural, 5 means no distortion and completely natural. I asked the subjects to rate from 1 to 5 how likely it was

that the speakers of each pair of audio clips could have been the same person, ignoring distortion, speed, and tone of speech, where one means completely different speakers and 5 represents the same person. The subjects didn't know if an audio clip was the ground truth or if it had been converted. The ratings which were not conclusive were excluded and thereby ended up with eight ratings for the analysis. All the raters were officially English speakers who were also students in NTNU. Since the StarGAN network was only tested on Data1 and Data2, we only show the results for Data2 and Data2 for both methods. The results are shown in the table below.

DATA	Method	Type	MOS
DATA 1	Ground Truth	M	4.61
		F	4.4
	StarGANv2-VC	M2F	4.3
	StarGAN-VC	M2F	2.15
DATA 2	Ground Truth	F	4.5
		M	4.45
	StarGANv2-VC	M2F	4.4
	StarGAN-VC	M2F	2.35

Table 4. 2: Mean Opinion score (MOS) for Data1 with Subjective metrics.

4.3.2 OBJECTIVE METRICS

I evaluated the result objectively by predicting the mean opinion score (pMOS) using MOSNET (Lo et al., 2019), which was computed using the python 3.6 environments. All the audios were saved in a file and loaded as a root directory to create the results as Mosnet raw txt by using the CNN_BLSTM model to build the MOSNET model and training using Cuda and applying librosa to extract the .wav file. I also reported the Character error rate (CER) using the ASR model discussed earlier for intelligibility. The results are shown below:

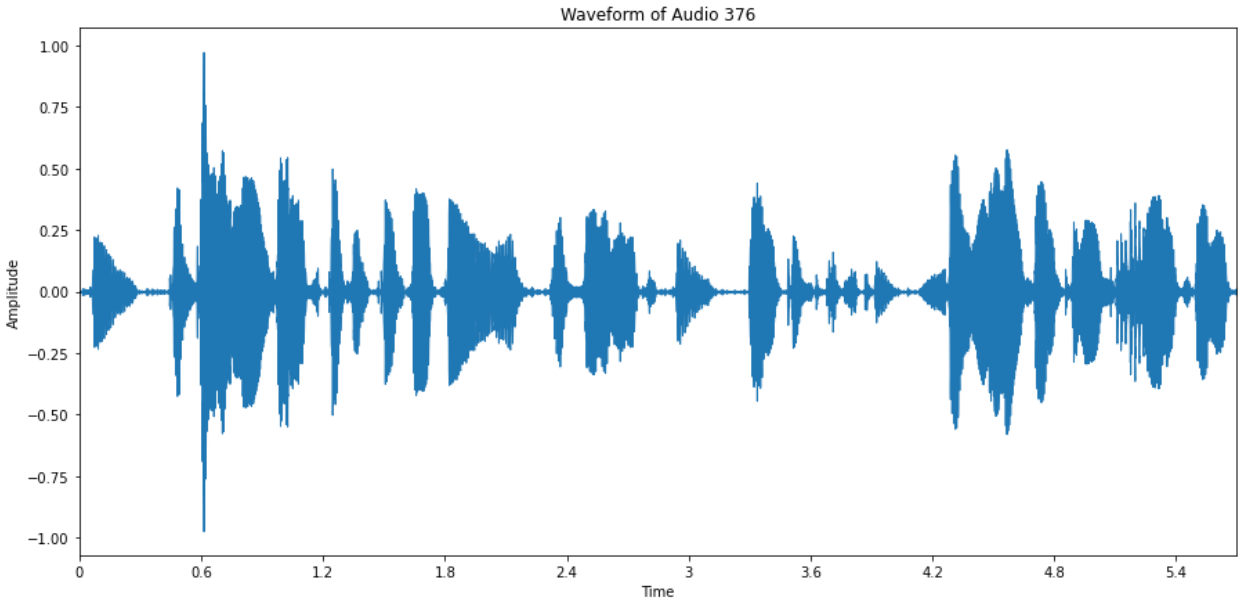
DATA	Method	pMOS	CER
DATA 1	Ground Truth	M-4.64	20.04%
		F-4.52	
	StarGANv2-VC	4.54	21.36%
	StarGAN-VC	2.10	50%
DATA 2	Ground Truth	F-4.70	18.07%
		M-4.40	
	StarGANv2-VC	4.65	19.34%
	StarGAN-VC	2.27	47%

Table 4. 3: Results showing the PMOS and CER using the objective metrics.

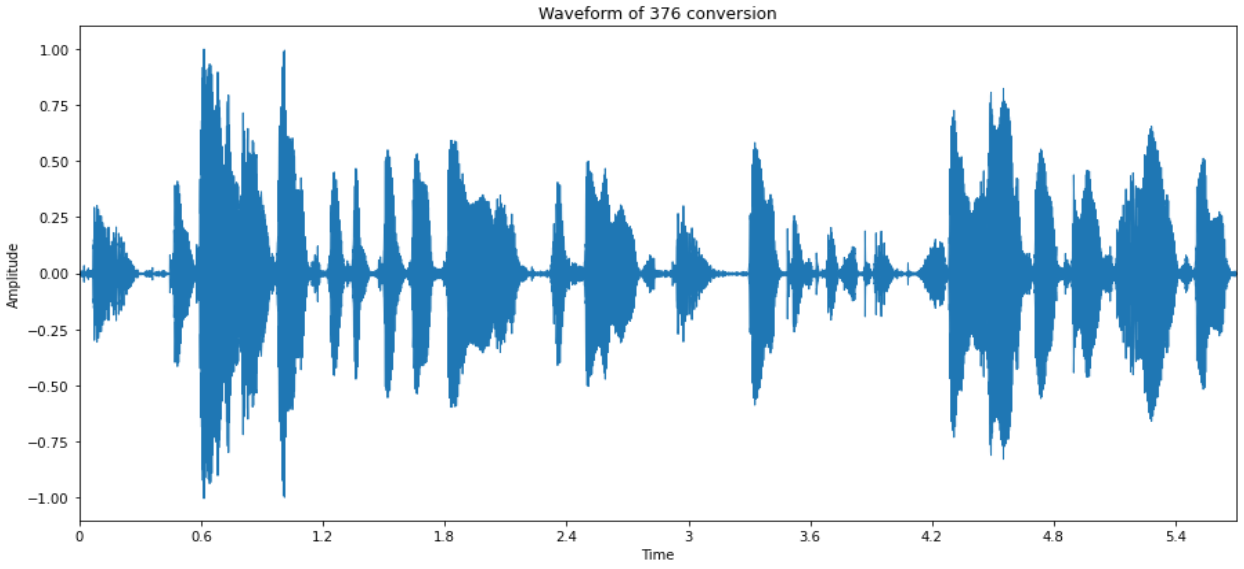
4.4 RESULTS

The results presented above from the experiments show that our method (StarGANv2-VC) outperforms the StarGAN-VC model in terms of naturalness and similarity. The accuracy of the speaker recognition model on the converted samples using our framework is much higher than the converted samples using the StarGAN-VC, and the predicted MOS (pMOS) of the converted pieces is significantly higher than that of StarGAN-VC. Finally, CER on the audio clips converted using our model is significantly lower than those converted using StarGAN-VC.

The framework is generalizable to a variety of voice conversion tasks with audios. The Melspectrogram of the samples converted vs. Ground truth is shown below, and the waveforms for Data1.

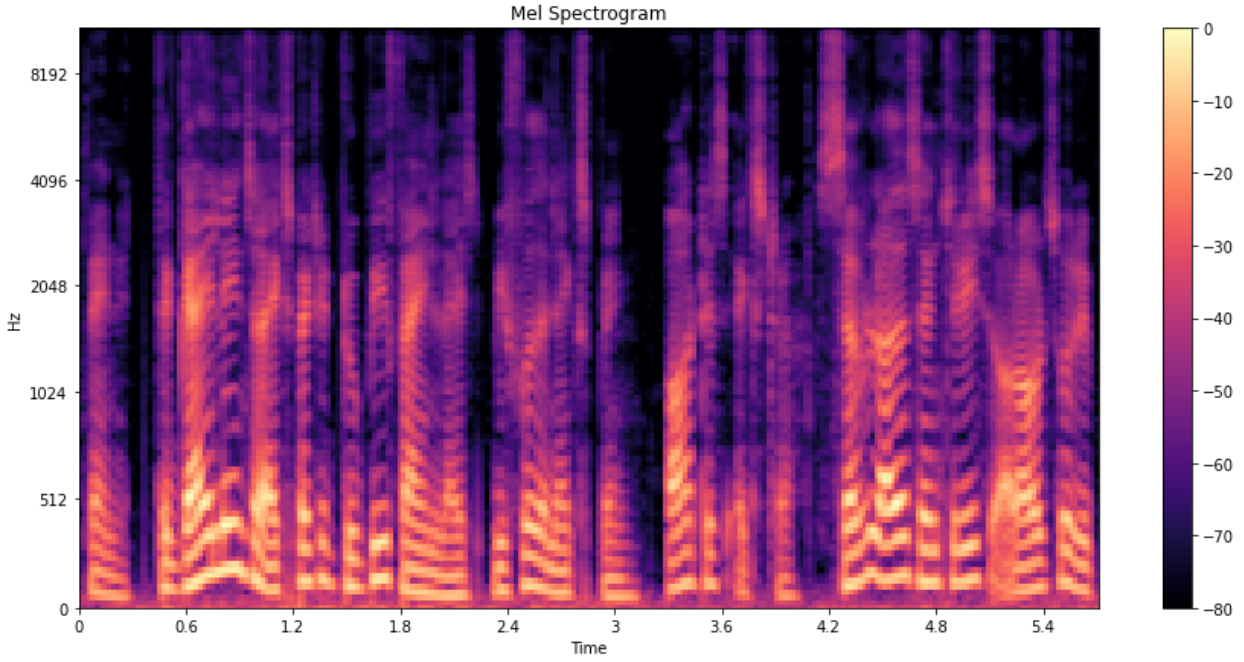


(a)

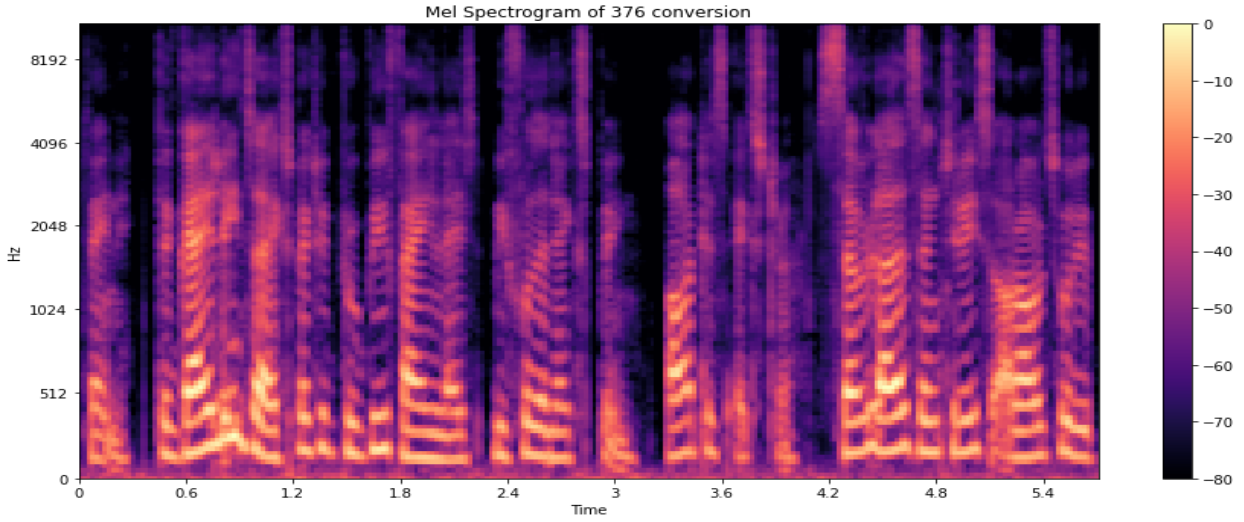


(b)

Figure 4. 1: Waveform of Audio signal for (a) Original audio of file before conversion for speaker P376 in Data 4 and (b) Converted audio of converted sample with P376 in Data 4 as source speaker.



(a)



(b)

Figure 4. 2: MelSpectrogram of Audio signal for (a) Original audio of file before conversion for speaker P376 in Data 4 and (b) Converted audio of converted sample with P376 in Data 4 as source speaker.

CHAPTER 5

5.0 DISCUSSION AND CONCLUSION

In this work, I presented an unsupervised framework with StarGANv2 for conversion with adversarial classifier and perpetual losses and F0 network and the Parallel WaveGAN Vocoder to achieve state-of-the-art nonparallel, cross-gender many-to-many voice conversion with outstanding performance in terms of naturalness and similarity.

This framework was also used to convert stylistic speech such as falsetto speech or emotional acting from plain reading source speech. It generalizes to several voice conversion tasks, such as any-to-many, cross-lingual, and singing conversion without the need for explicit training.

Based on the experiments carried out, this model seems to achieve a higher MOS score for conversion with regards to the previous model used for conversion, i.e., StarGAN-VC and is also faster in terms of training, it sounds more natural, and its similarity to the source speaker is far better and genuine.

5.1 FUTURE WORK

This work can be applied to cross-lingual voice conversion and to non-recognized languages like the Nigerian Pidgin and Nigerian Igbo language, which is a more robust work and contains lots of variety. Also, future work can include improving the quality of the converted samples with the StarGANv2 framework.

REFERENCES

Choi, Y., Uh, Y., Yoo, J., Ha, J.-W., 2020. StarGAN v2: Diverse Image Synthesis for Multiple Domains. arXiv:1912.01865 [cs].

Desai, S., Black, A.W., Yegnanarayana, B., Prahallad, K., 2010. Spectral Mapping Using Artificial Neural Networks for Voice Conversion. Trans. Audio, Speech and Lang. Proc. 18, 954–964. https://doi.org/10.1109/TASL.2010.2047683

Dhar, S., 2021. Introduction to Deep Learning-based Voice Conversion (VC): a growing domain of speech synthesis…. Medium. URL https://medium.com/@sandipandhar_6564/introduction-to-deep-learning-based-voice-conversion-vc-a-growing-domain-of-speech-synthesis-405ec7fa95b6 (accessed 12.15.21).

Ding, S., Gutierrez-Osuna, R., 2019. Group Latent Embedding for Vector Quantized Variational Autoencoder in Non-Parallel Voice Conversion, in: Interspeech 2019. Presented at the Interspeech 2019, ISCA, pp. 724–728. https://doi.org/10.21437/Interspeech.2019-1198

Doshi, K., 2021. Audio Deep Learning Made Simple: Automatic Speech Recognition (ASR), How it Works [WWW Document]. Medium. URL https://towardsdatascience.com/audio-deep-learning-made-simple-automatic-speech-recognition-asr-how-it-works-716cfce4c706 (accessed 12.13.21).

Garofolo, John S., Lamel, Lori F., Fisher, William M., Pallett, David S., Dahlgren, Nancy L., Zue, Victor, Fiscus, Jonathan G., 1993. TIMIT Acoustic-Phonetic Continuous Speech Corpus. https://doi.org/10.35111/17GK-BN40

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative Adversarial Networks.

Griffin, D., Lim, J.S., 1983. Signal estimation from modified short-time Fourier transform, in: ICASSP. https://doi.org/10.1109/ICASSP.1983.1172092

Gupta, S., Jaafar, J., wan Ahmad, W.F., Bansal, A., 2013. Feature Extraction Using Mfcc. SIPIJ 4, 101–108. https://doi.org/10.5121/sipij.2013.4408

Haykin, S., Lo, J., Fancourt, C., Príncipe, J., Katagiri, S., 2001. Nonlinear Dynamical Systems: Feedforward Neural Network Perspectives. undefined.

Imai, S., Sumita, K., Furuichi, C., 1983. Mel Log Spectrum Approximation (MLSA) filter for speech synthesis. Electronics and Communications in Japan (Part I: Communications) 66, 10–18. https://doi.org/10.1002/ecja.4400660203

Jain, L., Medsker, L., 1999. Recurrent Neural Networks: Design and Applications. https://doi.org/10.1201/9781420049176

Kameoka, H., Kaneko, T., Tanaka, K., Hojo, N., 2018. StarGAN-VC: Non-parallel many-to-many voice conversion with star generative adversarial networks. arXiv:1806.02169 [cs, eess, stat].

Kaneko, T., Kameoka, H., 2017. Parallel-Data-Free Voice Conversion Using Cycle-Consistent Adversarial Networks. arXiv:1711.11293 [cs, eess, stat].

Kaneko, T., Kameoka, H., Hiramatsu, K., Kashino, K., 2017a. Sequence-to-Sequence Voice Conversion with Similarity Metric Learned Using Generative Adversarial Networks, in: INTERSPEECH. https://doi.org/10.21437/INTERSPEECH.2017-970

Kaneko, T., Kameoka, H., Hojo, N., Ijima, Y., Hiramatsu, K., Kashino, K., 2017b. Generative adversarial network-based postfilter for statistical parametric speech synthesis, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Presented at the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4910–4914. https://doi.org/10.1109/ICASSP.2017.7953090

Kim, S., Hori, T., Watanabe, S., 2017. Joint CTC-Attention based End-to-End Speech Recognition using Multi-task Learning. arXiv:1609.06773 [cs].

Koolagudi, S.G., Rastogi, D., Rao, K.S., 2012. Identification of Language using Mel-Frequency Cepstral Coefficients (MFCC). Procedia Engineering, INTERNATIONAL CONFERENCE ON MODELLING OPTIMIZATION AND COMPUTING 38, 3391–3398. https://doi.org/10.1016/j.proeng.2012.06.392

Kum, S., Nam, J., 2019. Joint Detection and Classification of Singing Voice Melody Using Convolutional Recurrent Neural Networks. Applied Sciences 9. https://doi.org/10.3390/app9071324

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. https://doi.org/10.1038/nature14539

Li, Y.A., Zare, A., Mesgarani, N., 2021. StarGANv2-VC: A Diverse, Unsupervised, Non-Parallel Framework for Natural-Sounding Voice Conversion, in: Interspeech 2021. Presented at the Interspeech 2021, ISCA, pp. 1349–1353. https://doi.org/10.21437/Interspeech.2021-319

Lo, C.-C., Fu, S.-W., Huang, W.-C., Wang, X., Yamagishi, J., Tsao, Y., Wang, H.-M., 2019. MOSNet: Deep Learning based Objective Assessment for Voice Conversion. arXiv:1904.08352 [cs, eess].

Mohammadi, S.H., Kain, A., 2017. An overview of voice conversion systems. Speech Communication 88, 65–82. https://doi.org/10.1016/j.specom.2017.01.008

Mohammadi, S.H., Kain, A., 2014. Voice conversion using deep neural networks with speaker-independent pre-training: 2014 IEEE Workshop on Spoken Language Technology, SLT 2014. 2014 IEEE Workshop on Spoken Language Technology, SLT 2014 - Proceedings, 2014 IEEE Workshop on Spoken Language Technology, SLT 2014 - Proceedings 19–23. https://doi.org/10.1109/SLT.2014.7078543

Mouchtaris, A., Van der Spiegel, J., Mueller, P., 2004. Non-parallel training for voice conversion by maximum likelihood constrained adaptation, in: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing. Presented at the 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, p. I–1. https://doi.org/10.1109/ICASSP.2004.1325907

Moulines, E., Charpentier, F., 1990. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. Speech Communication 9, 453–467. https://doi.org/10.1016/0167-6393(90)90021-Z

Narendranath, M., Murthy, H.A., Rajendran, S., Yegnanarayana, B., 1995. Transformation of formants for voice conversion using artificial neural networks. Speech Communication, Voice Conversion: State of the Art and Perspectives 16, 207–216. https://doi.org/10.1016/0167-6393(94)00058-I

Ney, H., Suendermann, D., Bonafonte, A., Hoege, H., 2004. A first step towards text-independent voice conversion, in: Interspeech 2004. Presented at the Interspeech 2004, ISCA, pp. 1173–1176. https://doi.org/10.21437/Interspeech.2004-439

On, C.K., Pandiyan, P.M., Yaacob, S., Saudi, A., 2006. Mel-frequency cepstral coefficient analysis in speech recognition, in: 2006 International Conference on Computing & Informatics. Presented at the Informatics (ICOCI), IEEE, Kuala Lumpur, Malaysia, pp. 1–5. https://doi.org/10.1109/ICOCI.2006.5276486

Polyak, A., Wolf, L., Adi, Y., Taigman, Y., 2020. Unsupervised Cross-Domain Singing Voice Conversion. arXiv:2008.02830 [cs, eess].

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlıcˇek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K., 2011. The Kaldi Speech Recognition Toolkit 4.

Qian, K., Jin, Z., Hasegawa-Johnson, M., Mysore, G.J., 2020. F0-consistent many-to-many non-parallel voice conversion via conditional autoencoder. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 6284–6288. https://doi.org/10.1109/ICASSP40776.2020.9054734

Qian, K., Zhang, Y., Chang, S., Yang, X., Hasegawa-Johnson, M., 2019. AUTOVC: Zero-Shot Voice Style Transfer with Only Autoencoder Loss. arXiv:1905.05879 [cs, eess, stat].

Sisman, B., Yamagishi, J., King, S., Li, H., 2020. An Overview of Voice Conversion and its Challenges: From Statistical Modeling to Deep Learning. arXiv:2008.03648 [cs, eess].

Smith, J.. O., 2007. MATHEMATICS OF THE DISCRETE FOURIER TRANSFORM (DFT) WITH AUDIO APPLICATIONS SECOND EDITION [WWW Document]. URL https://ccrma.stanford.edu/~jos/st/ (accessed 12.27.21).

Stylianou, Y., 1996. Harmonic plus noise models for speech, combined with statistical methods, for speech and speaker modification. undefined.

Sun, L., Kang, S., Li, K., Meng, H., 2015. Voice Conversion Using Deep Bidirectional Long Short-Term Memory based Recurrent Neural Networks. https://doi.org/10.1109/ICASSP.2015.7178896

Sundermann, D., n.d. Text-Independent Voice Conversion 125.

Sundermann, D., Ney, H., Hoge, H., 2003. VTLN-based cross-language voice conversion, in: 2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No.03EX721). Presented at the 2003 IEEE Workshop on Automatic Speech Recognition and Understanding, IEEE, St Thomas, VI, USA, pp. 676–681. https://doi.org/10.1109/ASRU.2003.1318521

The Rainbow Passage | IDEA: International Dialects of English Archive [WWW Document], 2011. URL https://www.dialectsarchive.com/the-rainbow-passage (accessed 1.1.22).

Türk, O., 2007. CROSS-LINGUAL VOICE CONVERSION. undefined.

Watanabe, S., Hori, T., Karita, S., Hayashi, T., Nishitoba, J., Unno, Y., Soplin, N.E.Y., Heymann, J., Wiesner, M., Chen, N., Renduchintala, A., Ochiai, T., 2018. ESPnet: End-to-End Speech Processing Toolkit. arXiv:1804.00015 [cs].

Wikipedia, 2020. Word error rate. Wikipedia.

Yamamoto, R., Song, E., Kim, J.-M., 2020. Parallel Wavegan: A Fast Waveform Generation Model Based on Generative Adversarial Networks with Multi-Resolution Spectrogram, in: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Presented at the ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6199–6203. https://doi.org/10.1109/ICASSP40776.2020.9053795

Yi, Z., Zhang, H., Tan, P., Gong, M., 2017. DualGAN: Unsupervised Dual Learning for Image-to-Image Translation. 2017 IEEE International Conference on Computer Vision (ICCV). https://doi.org/10.1109/ICCV.2017.310

Zhang, X., Yao, J., He, Q., 2009. Research of STRAIGHT Spectrogram and Difference Subspace Algorithm for Speech Recognition, in: 2009 2nd International Congress on Image and Signal Processing. Presented at the 2009 2nd International Congress on Image and Signal Processing, pp. 1–4. https://doi.org/10.1109/CISP.2009.5303525

Zhu, J.-Y., Park, T., Isola, P., Efros, A., 2017. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. Arxiv.