Voice Conversion using GANs

A Project Report

submitted by

KOMMINENI ADITYA

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

June 2022

THESIS CERTIFICATE

This is to certify that the thesis titled **Voice Conversion using GANs**, submitted by **KOMMINENI ADITYA**, to the Indian Institute of Technology, Madras, for the award of credits for the courses **ID5490, ID5491 ID5492 : Dual Degree Project** in partial fulfilment of the requirements for the award of the degree of **Master of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Hema A Murthy
Research Guide
Professor
Dept. of Computer Science
IIT Madras, 600036
Place: Chennai
Date: 10th June 2022

ACKNOWLEDGEMENTS

First and foremost I am extremely grateful to my supervisor, Prof. Hema A Murthy for their invaluable advice, continuous support, and patience during my Dual Degree Project. Her immense knowledge and plentiful experience have encouraged me in all the time of my academic research and daily life.

I would like to express my gratitude to my parents. Without their tremendous understanding and encouragement in the past few months, it would be impossible for me to complete my study.

ABSTRACT

KEYWORDS: Generative Adversarial Networks; Voice Converison; Non Parallel; Indian Languages.

Speech processing related applications have grown to ubiquitous adoption in the past few years and have only been increased owing to the COVID-19 pandemic. Applications such as Automatic Speech Recognition, Text to Speech Synthesis, Speaker Identification and Voice Conversion are of particular interest. Although the former three have been successful in some capacity, voice conversion is one area which has still to see mainstream adoption. This is owing to the myriad of problems which one encounters while trying to perform voice conversion.

In this work, we will set baselines for voice conversion using Generative Adversarial Networks for Indian voices. We will propose some methods to improve the current Generative Adversarial Network based models and provide some analysis on the issues which hinder the performance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

GAN	Generative Adversarial Networks
VC	Voice Conversion
MCEP	Mel Cepstral Coefficients
MCD	Mel Cepstral Distortion
TTS	Text to Speech Synthesis

CHAPTER 1

INTRODUCTION

Voice conversion is formally defined as the process of converting a source speaker's utterance to the voice of a given target speaker without altering the linguistic contents of the utterance. The process of voice conversion mainly involves the conversion of two aspects of voice i.e. the prosody and the speaker characteristics(Yegnanarayana *et al.* (1984)). In this work, we will be focusing on speaker characteristics. Despite years of research, state of the art voice conversion models still exhibit deficiencies while trying to accurately mimic the target speaker prosodically, spectrally and at the same time maintaining high quality speech (Sisman *et al.* (2021)). Some major impairments for these performance degradations can be attributed to the fact that most models tend to over smooth the converted utterances which give an effect of a mix of both the source and target voice rather than that of the target voice alone(Hwang *et al.* (2013)).

In any voice conversion model, we have two phases namely, the training phase and the inference phase. A general training phase framework can be seen in Fig.1.1. During the training phase, we have two sets of utterances i.e. one from the source and one from the target speaker. These utterances are passed into the feature extractor. Models can use a variety of features such as the *LPC coefficients*(O'Shaughnessy (1988)), *Spectrograms*, *MCEP's*(Vergin *et al.* (1999)), *PPGs*(Phonetic Posteriograms)(Sun *et al.* (2016)) and so on. The most widely used feature in present day models is the MCEP. Once we extract the features, based on the model we want to train, we have an optional step wherein we align the two feature sets to account for the varying lengths. Once we have the aligned feature sets, we train the voice conversion model.

During the testing phase, we have the source speaker utterance which is passed through the feature extractor. Then, the features are converted to the target speaker's voice by the voice conversion model previously optimized in the training phase. A general framework for this phase is as seen in Fig.1.2.

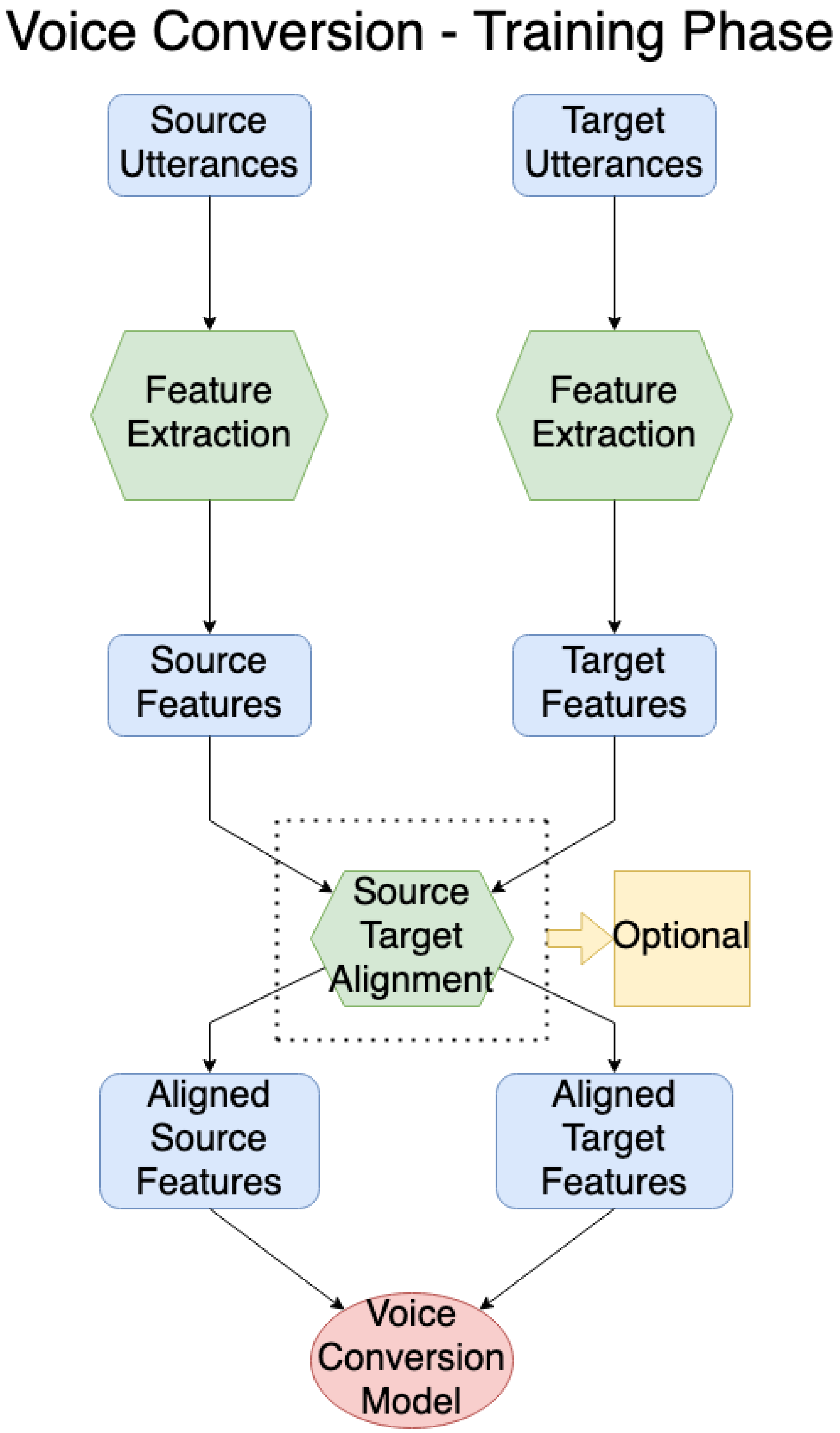

Figure 1.1: General Framework of a voice conversion model during the training phase.

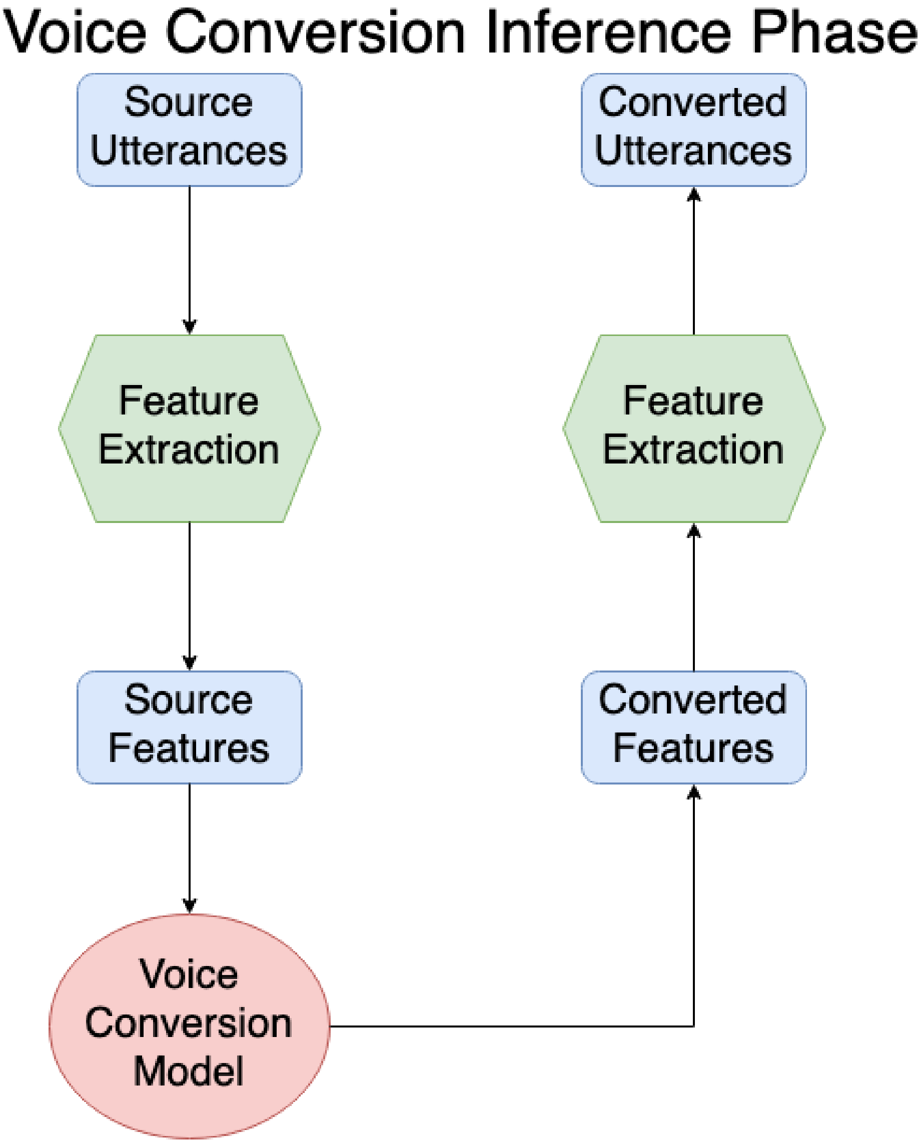

Figure 1.2: General Framework for Voice Conversion model in Inference/Test phase

The use cases of voice conversion are manifold. It can be used for personalizing TTS systems i.e. to obtain speech generation for new speakers with limited amount of training data and with lack of transcriptions. Voice conversion models are also used as a module in speech to speech translation systems wherein we would want to convert one language utterance to another language with the speaker characteristics intact. With the increasing demand for educational content in regional languages, voice conversion is extremely helpful to retain the voice of the instructor by first synthesizing the lecture in the desired language and then adapting to the desired voice. It can also be used in dubbing of movies, TV shows which greatly expedites the process, which otherwise is manually done. Additionally, voice conversion can sometimes be used to modify the voice and emotions of a given utterance so as to preserve privacy of the actual speaker.

Voice conversion systems can be classified on a variety of basis as seen in Fig.1.3. Based on the type of dataset available i.e. parallel dataset or non-parallel dataset. Parallel datasets are composed of pairs of utterances which have the same linguistic content with the only disparity being the speaker's voice. However, in the case of non-parallel, we have a set of sentences for each speaker without any common linguistic content which makes the problem of voice conversion much harder compared to the parallel case. In real world use cases, we almost always find the non-parallel case and seldom do we find the parallel case. Based on the language of the source and target speaker i.e. monolingual and crosslingual. If the source and target speaker utterances are in the same language, then it is a monolingual scenario whereas if the source and target utterances are in different languages, then it is a crosslingual scenario. Crosslingual voice conversion models are often based on unsupervised training as in Dines *et al.* (2013) or provide explicit linguistic information so as to allow the model to learn the linguistic mapping in addition to the voice mapping as seen in Zhou *et al.* (2019). Based on the number of speakers used to train the VC model i.e. one to one(Kaneko and Kameoka (2018)) or many to many(Kameoka *et al.* (2018)). One to one VC models consist of two speakers i.e. the source speaker and target speaker. The objective of the model is to learn the mapping which converts the source utterance to the target utterance. In case of many to many VC, we have a set of speakers and we want to have the ability to convert a given source speaker's utterance into any of the

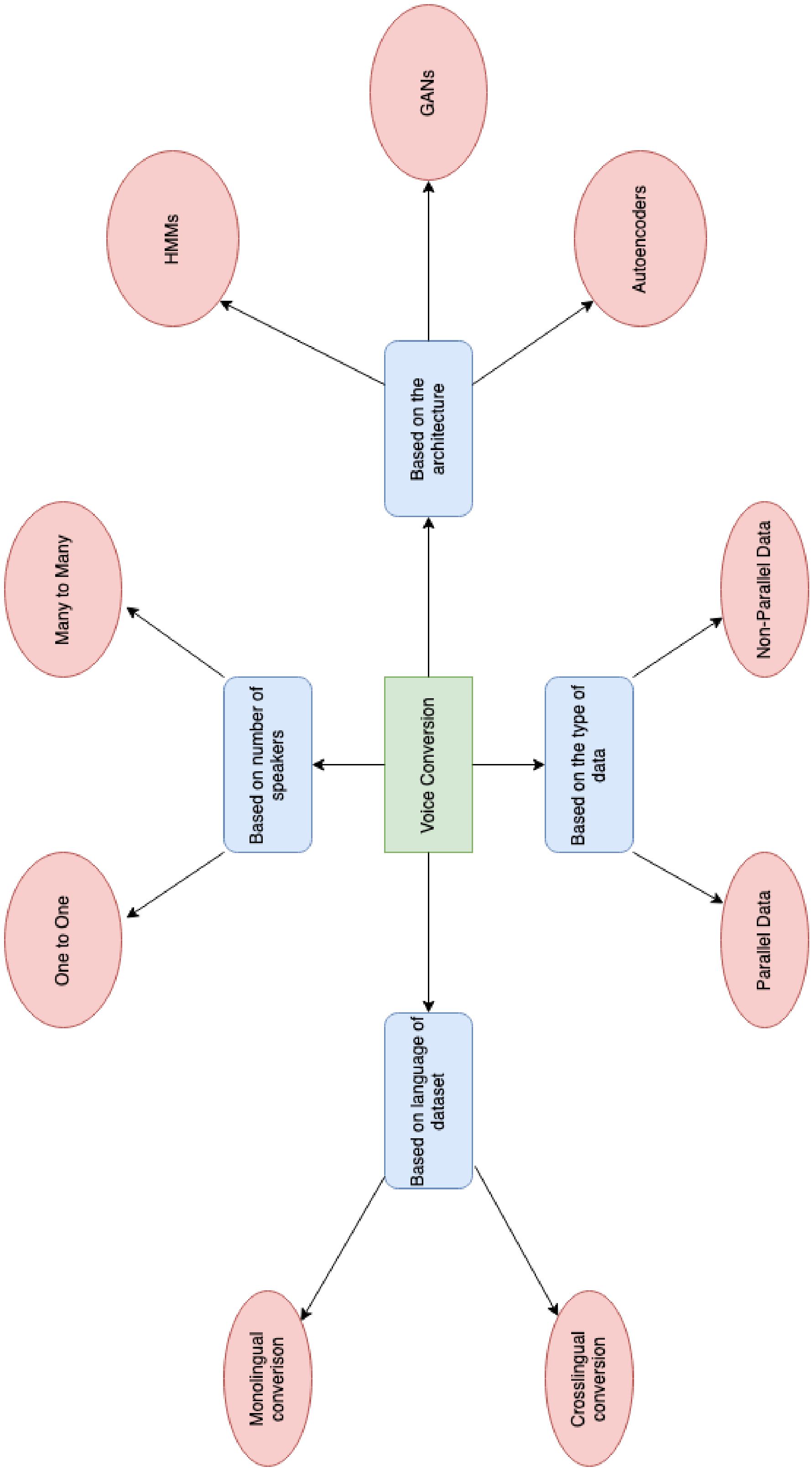

Figure 1.3: Classification of voice conversion models based on various factors i.e. dataset type, language of the various speakers, number of speakers and the architecture of the voice conversion model

other speakers in the set. Since the number of mappings which need to be learnt in many to many VC is much larger compared to a one to one case, it requires substantially more data compared to the one to one case. Based on the usage of transcripts, VC models are classified as text dependent voice conversion and text independent voice conversion. Text independent voice conversion models are preferred owing to their independence on transcriptions.

1.1 Dataset

For all the experiments in this thesis, IndicTTS dataset(Baby *et al.* (2016)) has been used. It is composed of native Indian language and Indian English utterances from speakers belonging to various nativities. Table.1.1 consists of the information about the various speakers and the corresponding time duration. From the Table.1.1, the first column refers to the specific speaker dataset i.e. the first word refers to the native language the speaker speaks, the second word is the gender of the speaker and the third word is the language in which that specific set of utterances are recorded in. So, *Hindi Female Hindi* refers to a female native Hindi speaker and the utterances are recorded in Hindi. For testing, a set of parallel utterances were extracted from all the utterances in IndicTTS dataset. All the utterances were resampled to 16kHz and were normalized prior to training of the models.

1.2 Evaluation Metrics

The types of objective evaluation metrics which can be used to quantify the performance of a voice conversion model have not been extremely clear. Here, we will be using MCD and x-vector based evaluation metrics to quantify the performance of a voice conversion model. MCD can be considered to be measuring the naturalness of the utterances whereas the X-vector based metrics act as a measure for speaker similarity.

1.2.1 Mel Cepstral Distortion (MCD)

MCD(Kubichek (1993)) scores can be computed directly on datasets wherein there is some portion of parallel utterances. Mel Cepstral Distortion is defined as the process of computing the difference between the target utterance cepstra and the voice converted utterance cepstra. The formula for the same is shown below:

$$MCD(v^{Source}, v^{Target}) = \frac{\alpha}{T}\Sigma_{t=0}^{T-1}\sqrt{\Sigma_{d=s}^{D}(v_d^{Source}(t) - v_d^{Target}(t))^2}$$

$$\alpha = \frac{10\sqrt{2}}{\log 10} = 6.148$$

In the equation above, v^{Source}, v^{Target} refer to the time aligned cepstral coefficients of source and target utterance respectively. For non-parallel test corpora, we cannot use MCD as an evaluation metric. The lower the MCD value, the closer the two utterances are in terms of similarity in spectrum. MCD as a metric for voice conversion can help us identify the degree to which the spectrum of the converted utterance matches with that of the target utterance. MCD is useful as a metric to determine the naturalness of a given converted utterance rather than the speaker similarity. This is owing to the fact that the cepstra will have both speaker information and spectral information. However, it doesn't explicitly account for speaker information. Hence, a lower MCD would generally indicate a better naturalness.

1.2.2 X-vectors

X-vectors(Snyder *et al.* (2018)) were primarily proposed in order to improve the speaker identification task by producing speaker embeddings which were robust to language changes, noise and prosody. Here, we would like to use these speaker embeddings in order to evaluate the performance of the voice conversion system.

X-vector Cosine Similarity

After computing the x-vectors, we will compute the inner product of each of the converted utterances with respect to the target utterance x-vectors and take the

Dataset	Duration	Average Duration	Number of Utterances
Hindi Male English	709	2.36	300
Tamil Male English	716	2.38	300
Hindi Female English	715	2.38	300
Tamil Female English	713	2.37	300
Hindi Male Hindi	712	2.37	300
Hindi Female Hindi	713	2.37	300
Tamil Male Tamil	715	2.38	300
Tamil Female Tamil	716	2.38	300
Telugu Male Telugu	707	2.35	300
Telugu Female Telugu	1751	5.88	300
Telugu Male English	1241	4.14	300
Telugu Female English	1677	5.59	300

Table 1.1: Statistics of dataset

average. If the voice conversion model was ideal, we would get a cosine similarity close to 1 for the same speakers and for dissimilar speakers, we would get a cosine similarity close to 0. Based on these values, we can determine how well the model is able to convert one speaker to the other.

X-vector tSNE Plots

T-distributed stochastic neighbour(van der Maaten and Hinton (2008)) embedding is a statistical method to visualize higher dimensional data in a 2 or 3 dimensional plot. This is a non linear dimensionality reduction technique wherein two points in the high dimensional space close to each other are represented in the 2 or 3 dimensional space also closeby. Therefore, this gives us an idea of the clusters which are formed in the higher dimensional space. Here, in terms of voice conversion, by plotting the t-SNE plots for the x-vectors of the target and converted voices, we can analyse on how well the conversion is being performed and determine if the model is converting one voice to the other or if it is averaging both the voices i.e. retaining some characteristics of the source voice.

CHAPTER 2

Voice Conversion using CycleGAN

GANs(Goodfellow *et al.* (2014)) were introduced as deep generative models which had the capability to model distributions and generate samples post training which was an obstacle conventional neural networks couldn't tackle. Following this, an image to image translation network called CycleGAN(Zhu *et al.* (2017)) was proposed which could convert one art style to the other. Now, one could redefine voice conversion as the process of converting one style to the other with the style here being the voice of the respective speaker. Based on this definition, CycleGAN-VC(Kaneko and Kameoka (2018)) was one proposed GAN model to perform voice conversion.

2.1 Architecture

CycleGAN is a non-parallel, one to one voice conversion model. It doesn't require the transcriptions for utterances in order to convert the voice of one speaker to another.

CycleGAN converts the pitch of a source speaker to the target speaker using a simple linear transform. The expression for the same is as follows:

$$f0_y = \frac{(f0_x - \mu_x)\sigma_y}{\sigma_x} + \mu_y$$

Let us consider the two speakers to be X and Y. Each CycleGAN has two generators and two discriminators i.e. one generator to model distribution Y with inputs as X and vice versa. The discriminators on the other hand try to determine if the input given belongs to the actual distribution or is an output from the generator. Architecture of the generator can be as seen in Fig.2.1 and the discriminator can be as seen in Fig.2.2.

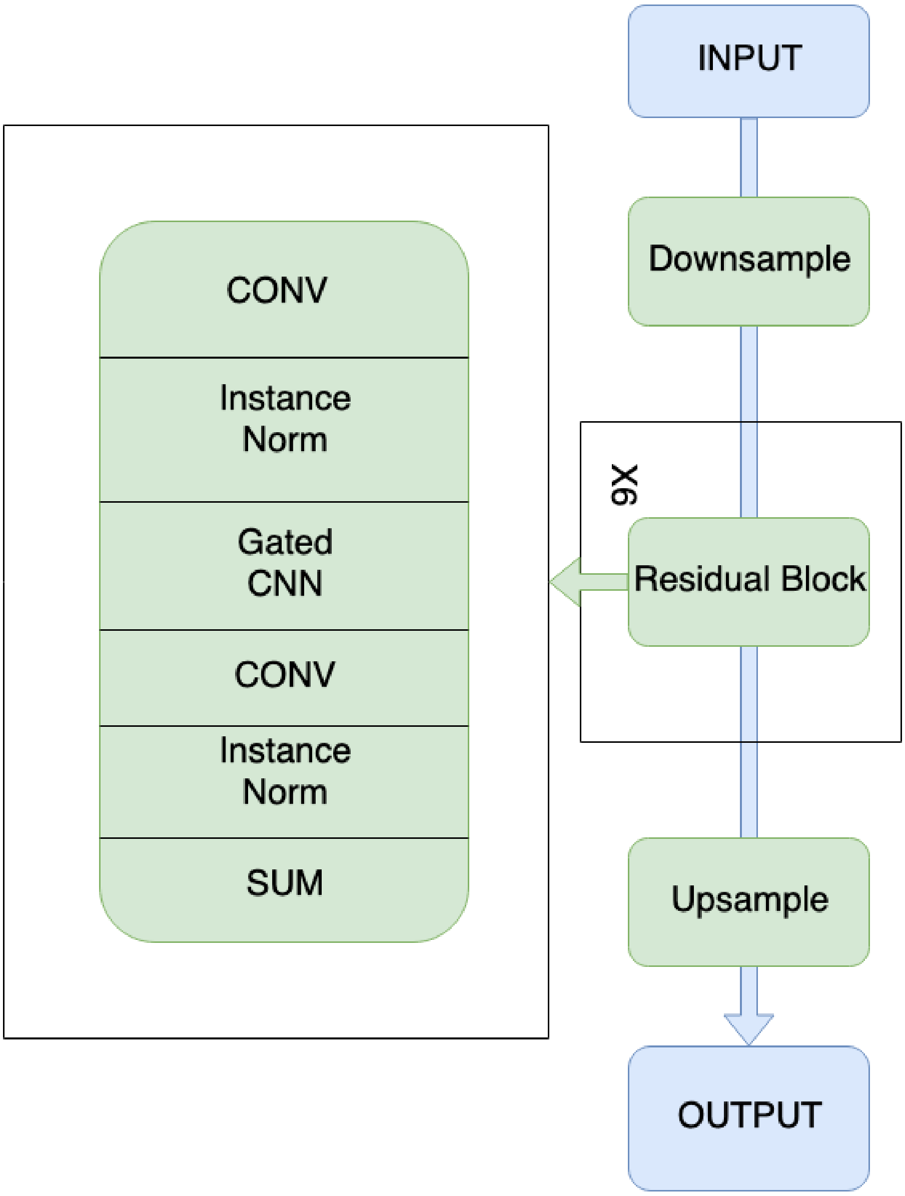

Figure 2.1: Architecture of a Generator in CycleGAN

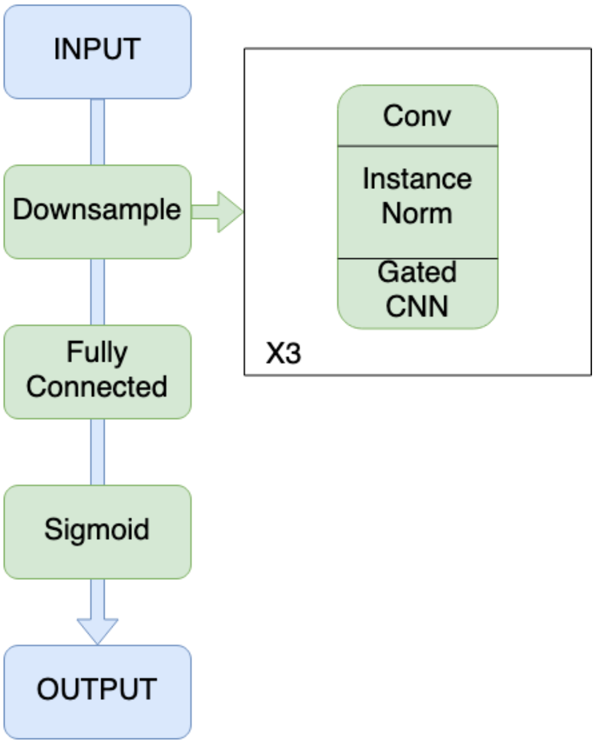

Figure 2.2: Architecture of a Discriminator in CycleGAN

2.1.1 Gated CNN

As we can see in Fig.2.1, the generator of a CycleGAN is composed of a variety of layers among which one of them is a GatedCNN(Dauphin *et al.* (2017*a*)). We are aware of the fact that speech consists of both sequential and hierarchical components. In order to model the sequential components, we need to have layers which are able to account for the neighbourhood such as RNN'sMedsker and Jain (2001), LSTM's(Hochreiter and Schmidhuber (1997)) or GLU's(Dauphin *et al.* (2017*b*)). However, these components are computationally expensive and cannot be parallelized for training. Hence, CycleGAN makes use of GatedCNN which allows for parallelism and at the same time can model some sequential aspects i.e. segmental features. The expression for the Gated CNN is as shown below:

$$h(X) = (X * W + b) \bigotimes \sigma(X * V + c)$$

Where X is the input; W, V, b and c are the weights of the layer. Although these layers do not have neighbourhood information beyond the present frame, they are able to model sequential information within the frame which helps in better modelling the voice.

2.1.2 Loss Function

The combined objective i.e. loss function of CycleGAN is as given below:

$$L_{total} = L_{adv}(G_{X->Y}, D_Y) + L_{adv}(G_{Y->X}, D_X) + \lambda_{cyc} L_{cyc}(G_{X->Y}, G_{Y->X})$$

In the above equation, L_{adv} is the adversarial loss, L_{cyc} is the cyclo consistent loss and λ_{cyc} is a hyperparameter to determine the relative importance of both the losses.

Adversarial Loss

The adversarial loss plays an important role in the training of all GAN models. The adversarial loss plays a min-max optimization role wherein the discriminator will try to minimize the loss i.e. increase the probability with which the discriminator is correctly able to predict the actual datapoints from that generated by the generator. Whereas the generator will try to maximize the loss wherein it tried to evade the detection of the discriminator and pass as actual datapoints. This loss helps the generator to learn the characteristics of the distribution it needs to model. The expression for the loss is as shown below:

$$L_{adv}(G_{X->Y}, D_Y) = E_{y \sim P_{Data}(y)}[\log D_Y(y)] + E_{x \sim P_{Data}(x)}[\log(1 - D_Y(G_{X->Y}(x))]$$

However, without imposing any specific constraints, the model will not be able to preserve the linguistic content and will not be able to converge to an optimal set of parameters.

Cyclo Consistent Loss

In addition to the GAN model learning the distribution of the target speakers voice characteristics, we would also want the model to preserve the linguistic content of the corresponding utterance at the same time. In order to incorporate this, adversarial loss alone is not sufficient. Therefore, we have the cycle-consistency loss whose expression is as given below:

$$L_{cyc} = E_{x \sim P_{Data}(x)}[||G_{Y->X}(G_{X->Y}(x)) - x||_1] + E_{y \sim P_{Data}(y)}[||G_{X->Y}(G_{Y->X}(y)) - y||_1]$$

As we can see from the expression, the objective of this loss term is to ensure that when a given utterance has been passed through both the generators i.e. $G_{X->Y}$ and $G_{Y->X}$, the L1 distance between the output obtained and the original utterance is as minimal as possible. This would help the model preserve the linguistic content.

Identity Loss

Despite the inclusion of the Cyclo consistency loss parameter, in practice it was found that the model was still not able to completely preserve the linguistic information. In order to further constrain the model, Identity loss was introduced whose expression is as seen below:

$$L_{id} = E_{x \sim P_{Data}(x)}[||G_{Y->X}(x) - x||_1] + E_{y \sim P_{Data}(y)}[||G_{X->Y}(y) - y||_1]$$

In the image to image translation papers as in Zhu *et al.* (2017), this loss parameter was said to account for colour preservation. From the expression, we can see that this establishes a constraint on the model to minimize the distance between the target utterance Y and the output of $G_{X->Y}$ while passing target utterance Y. This loss parameter is used only for a few iterations during the beginning and is then set to zero for the rest of the training.

2.2 Implementation Details

All the utterances were sampled at 16kHz and normalized. The CycleGAN uses MCEP features as inputs. We use the WORLD package in order to compute the fundamental frequency, spectral features and aperiodicities. The spectral features are then converted to MCEP's.

For all the experiments which follow, adam optimizer is used with paramemters

	HinM/TamM-HinM	TamM/HinM-TamM
MCD	7.486	8.267

Table 2.1: MCD scores for HinM-TamM experiment. In the table above, TamM-HinM refers to the utterances which have been converted from Tamil speakers voice to Hindi speakers voice.

Speakers	Hin M	Tam M
Tam M - Hin M	0.698	0.632
Hin M - Tam M	0.612	0.638

Table 2.2: Cosine Similarity Scores for HinM-TamM

β_1 and β_2 as 0.5 and 0.999 respectively. Unless mentioned, the number of MCEP feautures used are 24. The frame period per speech frame is 5ms and the number of speech frames used in one window of CycleGAN is 128. The values of λ_{cyc} and λ_{id} are 10 and 5 respectively. λ_{id} is set to zero after the first $1e^4$ iterations. The generator and discriminator learning rates are linearly decayed after the first $2e^5$ iterations.

2.3 Experiments on Indian English

Firstly, we trained models on Indian speakers with training utterances in English. The experiments performed are named as HinM-TamM, HinM-TamF, HinF-TamF, HinF-TamM. For example, HinM-TamF refers to the training of CycleGAN wherein the two speakers are Hindi native male speaker and Tamil native female speaker. This training set gives us an exhaustive set of results in the possible combinations of genders i.e. male-male, male-female, female-female and female-male. The results for the same are presented in the following subsections.

2.3.1 HinM-TamM

The MCD values for this experiment set is as shown in Tab.2.1 and the x-vector cosine similarity score are as seen in Tab.2.2. X-vector based tSNE plot for the same is as seen in Fig.2.3. From the cosine similarity scores and tSNE plot, we can see that the model is not able to convert the voice from one to other extremely well.

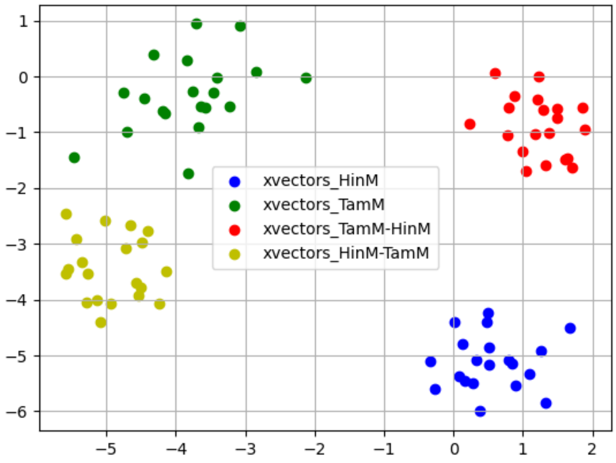

Figure 2.3: tSNE plot for HinM-TamM experiment.

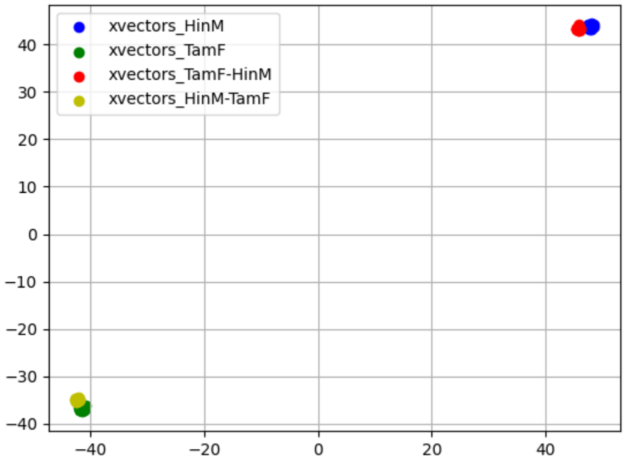

Figure 2.4: tSNE plot for HinM-TamF experiment.

2.3.2 HinM-TamF

The MCD values for this experiment set is as shown in Tab.2.3 and the x-vector cosine similarity score are as seen in Tab.2.4. X-vector based tSNE plot for the same is as seen in Fig.2.4. From the tSNE plot and cosine similarity scores, we can see that the model is able to perform better compared to the male-male scenario.

	HinM/TamF-HinM	TamF/HinM-TamF
MCD	10.408	7.142

Table 2.3: MCD scores for HinM-TamF experiment. In the table above, TamM-HinF refers to the utterances which have been converted from Tamil speakers voice to Hindi speakers voice.

Speakers	Hin M	Tam F
Tam F - Hin M	0.675	-0.03
Hin M - Tam F	0.026	0.688

Table 2.4: Cosine Similarity Scores for HinM-TamF

2.3.3 HinF-TamF

The MCD values for this experiment set is as shown in Tab.2.6 and the x-vector cosine similarity score are as seen in Tab.2.5. X-vector based tSNE plot for the same is as seen in Fig.2.5. The performance of the model is similar to the male-male case wherein it is unable to perform well.

2.3.4 HinF-TamM

The MCD values for this experiment set is as shown in Tab.2.8 and the x-vector cosine similarity score are as seen in Tab.2.7. X-vector based tSNE plot for the same is as seen in Fig.2.6. The model performs well and is similar to the case of HinM-TamF.

2.3.5 Observations

- From the experiments above, we can clearly see that CycleGAN is able to perform better when there is conversion between the genders i.e. the two speakers belong to the different gender.
- In the tSNE plots, we can see that in the case of HinM-TamM and HinF-TamF, the points are scattered midway between both the actual speaker points. This points to the fact that the features could be extremely similar

Speakers	Hin F	Tam F
Tam F - Hin F	0.73	0.68
Hin F - Tam F	0.737	0.702

Table 2.5: Cosine Similarity Scores for HinF-TamF

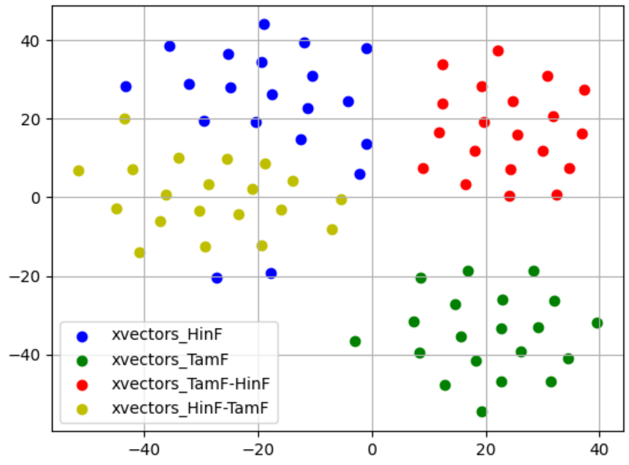

Figure 2.5: tSNE plot for HinF-TamF experiment.

	HinF/TamF-HinF	TamF/HinF-TamF
MCD	10.282	9.05

Table 2.6: MCD scores for HinF-TamF experiment. In the table above, TamF-HinF refers to the utterances which have been converted from Tamil speakers voice to Hindi speakers voice.

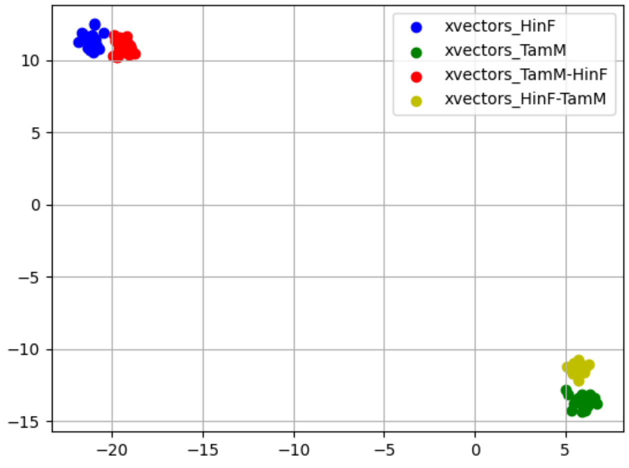

Figure 2.6: tSNE plot for HinF-TamM experiment.

Speakers	Hin F	Tam M
Tam M - Hin F	0.74	0.01
Hin F - Tam M	0.09	0.575

Table 2.7: Cosine Similarity Scores for HinF-TamM

	HinF/TamM-HinF	TamM/HinF-TamM
MCD	8.03	9.75

Table 2.8: MCD scores for HinF-TamM experiment. In the table above, TamM-HinF refers to the utterances which have been converted from Tamil speakers voice to Hindi speakers voice.

	HinM/HinF-HinM	HinF/HinM-HinF
English	10.211	8.43
Hindi	9.21	7.903

Table 2.9: MCD scores for HinM- HinF experiment for both english and Hindi. In the table above, HinM-HinF refers to the utterances which have been converted from Hindi male speakers voice to Hindi female speakers voice.

to each other and the CycleGAN model is unable to find the mapping of one speaker to the other.

- From the MCD tables, we can see that when the target speaker is male, the MCD values are consistently lower than that when the target speaker is female.

2.4 Experiments on Native Indian Languages

In these set of experiments, we will pick a male and female speaker from each of Hindi, Tamil and Telugu languages. For each of the pair of speakers, we will train an Indian English and native language CycleGAN model. This will help us determine if the model is able to perform equally well irrespective of language or if there are some difficulties with specific languages. Ideally, we would expect a language agnostic behaviour.

2.4.1 Hindi

From the Tab.2.9, we can see the MCD scores for both the English and Hindi experiments. The x-vector based cosine similarity plots for the same are as seen in Tab.2.10 and Tab.2.11 for English and Hindi respectively. The figure for the tSNE plot of the experiment can be seen in Fig.2.7

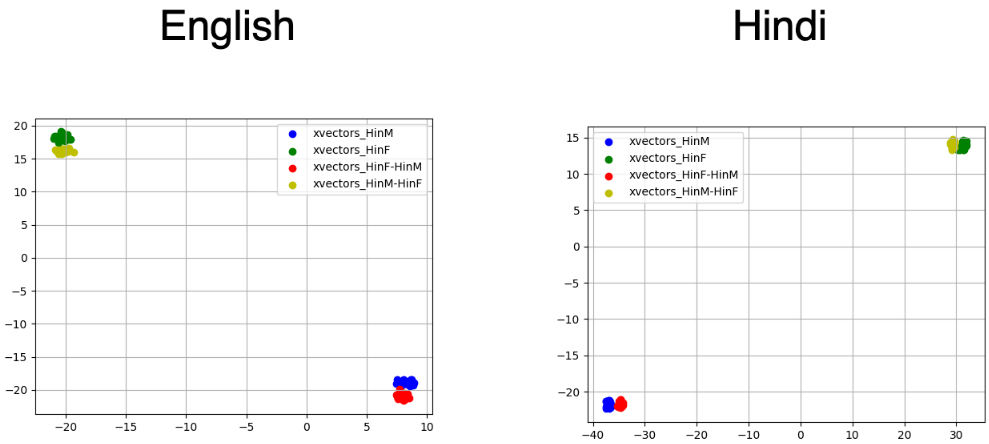

Figure 2.7: tSNE plots for the HinM-HinF experiment, The plot on the left is the English dataset plot and the one on the right is the Hindi dataset plot.

Speakers	Hin M	Hin F
Hin F - Hin M	0.692	0.03
Hin M - Hin F	0.01	0.725

Table 2.10: Cosine Similarity Scores for HinM-HinF English

Speakers	Hin M	Hin F
Hin F - Hin M	0.63	0.142
Hin M - Hin F	0.07	0.703

Table 2.11: Cosine Similarity Scores for HinM-HinF Hindi

	TamM/TamF-TamM	TamF/TamM-TamF
English	9.9	7.26
Tamil	10.4	7.73

Table 2.12: MCD scores for TamM- TamF experiment for both english and tamil. In the table above, TamM-TamF refers to the utterances which have been converted from Tamil male speakers voice to Tamil female speakers voice.

Speakers	Tam M	Tam F
Tam F - Tam M	0.59	0.06
Tam M - Tam F	0.05	0.67

Table 2.13: Cosine Similarity Scores for TamM-TamF English

2.4.2 Tamil

From the Tab.2.12, we can see the MCD scores for both the English and Tamil experiments. The x-vector based cosine similarity plots for the same are as seen in Tab.2.13 and Tab.2.14 for English and Tamil respectively.

2.4.3 Telugu

From the Tab.2.15, we can see the MCD scores for both the English and Telugu experiments. The x-vector based cosine similarity plots for the same are as seen in Tab.2.16 and Tab.2.17 for English and Telugu respectively.

2.4.4 Observations

- Firstly, we must note that the Indian language experiments perform better than the English experiments in terms of MCD.
- However, when we look at the cosine similarity scores, Indian language experiments perform slightly poorer compared to their English counterparts.
- These experiments prove the point that the performance of both the Indian languages and English is not extremely far apart. Hence, we can arrive at the conclusion that the CycleGAN is language agnostic.

Speakers	Tam M	Tam F
Tam F - Tam M	0.69	0.07
Tam M - Tam F	0.02	0.68

Table 2.14: Cosine Similarity Scores for TamM-TamF Tamil

	TelM/TelF-TelM	TelF/TelM-TelF
English	11.55	10.7
Telugu	8.49	8.63

Table 2.15: MCD scores for TelM- TelF experiment for both english and tamil. In the table above, TelM-TelF refers to the utterances which have been converted from Telugu male speakers voice to Telugu female speakers voice.

Speakers	Tel M	Tel F
Tel F - Tel M	0.646	0.17
Tel M - Tel F	0.29	0.75

Table 2.16: Cosine Similarity Scores for TelM-TelF English

2.5 Experiment on Crosslingual Voice Conversion

Although CycleGAN doesn't provide any explicit mechanism for modelling the language information, we try out an experiment with Hindi male and Telugu Female speaking their respective native languages i.e. Hindi and Telugu respectively. This would be crosslingual voice conversion. For this experiment, we have tried two variations. One of the experiments being the vanilla experiment with no changes to the input features. The other being the case wherein some of the input features were masked. Considering that the datasets are cross lingual, we will not be able to compute the MCD since we don't have parallel corpora.

2.5.1 Vanilla Features

The cosine similarity scores for the experiment with 24 MCEP features can be seen in Tab.2.18.

Speakers	Tel M	Tel F
Tel F - Tel M	0.587	0.203
Tel M - Tel F	0.107	0.727

Table 2.17: Cosine Similarity Scores for TelM-TelF Telugu

Speakers	Hin M	Tel F
Tel F - Hin M	0.653	0.005
Hin M - Tel F	0.09	0.722

Table 2.18: Cosine Similarity Scores for HinM-TelF Cross lingual vanilla features

Speakers	Hin M	Tel F
Tel F - Hin M	0.42	0.3
Hin M - Tel F	0.28	0.35

Table 2.19: Cosine Similarity Scores for HinM-TelF Cross lingual masked features

2.5.2 Masked Features

In the MCEP features, the higher coefficients are responsible for the speaker characteristics whereas the lower coefficients are responsible for containing the linguistic information. Hence, since we want to preserve the linguistic information of the utterance we set the first 8 coefficients of the MCEP to zero and train the model. During inference, we convert the unmasked MCEP coefficients using the trained model and pass the maked coefficients without any transform. The cosine similarity scores for this experiment are as seen in Tab.2.19.

2.5.3 Observations

- From the cosine similarity scores of the vanilla crosslingual conversion, we can see that the cosine similarity is being established.
- Although the outputs of the masked features do not have high speaker similarity scores, we see that these outputs do not have as much noise. Hence, if we are able to obtain source content seperation, we can perform better cross lingual conversion with CycleGAN.
- We can conclude that CycleGAN can be used. for crosslingual voice conversion. However, there are some artefacts and noise introduced into the converted audio files which are undesirable. This working could be attributed to the residual layers which better help the model select the dimensions it wants to keep unchanged and the ones which need to be changed.

	HinM/TamM-HinM	TamM/HinM-TamM
MCD	7.83	7.31

Table 2.20: MCD scores for HinM-TamM Histogram Cascade experiment using CycleGAN.

Speakers	Hin M	Tam M
Tam M - Hin M	0.711	0.602
Hin M - Tam M	0.621	0.634

Table 2.21: Cosine Similarity Scores for HinM-TamM Histogram Equalization cascade with CycleGAN

2.6 Experiment on cascaded Histogram Equalization

Histogram Equalization(Zhihong and Xiaohong (2011)) is an image processing technique primarily employed in order to enhance the contrast of a given image. This has been shown to improve the performance of HTS based text to speech synthesis(Murthy *et al.* (2020)). Here, we would want to verify if the cascade of CycleGAN followed by Histogram Equalization can improve the voice conversion performance. For this, we will perform the experiments HinM-TamM and HinM-TamF both in Indian English.

2.6.1 HinM-TamM

The MCD scores for HinM-TamM can be seen in Tab.2.20. Cosine similarity scores of x-vectors can be seen in Tab.2.21.

2.6.2 HinM-TamF

The MCD scores for HinM-TamM can be seen in Tab.2.22. Cosine similarity scores of x-vectors can be seen in Tab.2.23.

	HinM/TamF-HinM	TamF/HinM-TamF
MCD	7.1	9.9

Table 2.22: MCD scores for HinM-TamF Histogram Cascade experiment using CycleGAN.

Speakers	Hin M	Tam F
Tam F - Hin M	0.66	0.02
Hin M - Tam F	0.05	0.70

Table 2.23: Cosine Similarity Scores for HinM-TamF Histogram Equalization cascade with CycleGAN

	HinM/TamF-HinM	TamF/HinM-TamF
12 MCEP	7.345	10.59
24 MCEP	7.142	10.4
40 MCEP	7.02	10.1

Table 2.24: MCD scores for HinM-TamF for varying MCEP values.

2.6.3 Observations

- From the plots, we can see that the Histogram Equalization cascade has lesser MCD values compared to using the CycleGAN alone.
- With respect to the cosine similarity scores, we do not see any perceivable improvement when cascading with histogram equalization.
- We can conclude that the usage of histogram equalization can help reduce the amount of noise present in the converted utterance. However, in terms of improving the speaker similarity, Histogram Equalization doesn't help.

2.7 Experiment on the number of MCEP Features

In this experiment, we vary the number of MCEP features used during the training of the CycleGAN to determine the optimal value of the number of MCEP features. Here, we will use 12 and 40 MCEP feature numbers in order to compare it to 24 MCEP features. The speakers chosen for this experiment are HinM and TamF English dataset. The MCD scores for the various MCEP Features are as shown in the Tab.2.24. The cosine similarity scores for 12 MCEP and 40 MCEP are as shown in Tab2.25 and Tab.2.26 respectively.

From the MCD values, we can see that as the number of MCEP features are increased, the MCD values decrease. Also, we see that the cosine similarity scores

Speakers	Hin M	Tam F
Tam F - Hin M	0.626	0.03
Hin M - Tam F	0.11	0.6

Table 2.25: Cosine Similarity Scores for HinM-TamF for 12 MCEP.

Speakers	Hin M	Tam F
Tam F - Hin M	0.68	0.06
Hin M - Tam F	0.01	0.72

Table 2.26: Cosine Similarity Scores for HinM-TamF for 40 MCEP.

are not affected to a large extent with the MCD scores. Hence, the number of MCEP features doesn't majorly affect the speaker similarity.

2.8 Experiment on using LPC features

Until now, we used MCEP features in order to train the CycleGAN. However, we know that MCEP doesn't model the pitch extremely well which is important for tasks such as voice conversion. Hence, CycleGAN model was trained using LSF(Kabal and Ramachandran (1986)) coefficients and the corresponding residual.

2.8.1 Procedure

Firstly, we compute the LPC(O'Shaughnessy (1988)) coefficients and residual for the dataset utterances. Following this, we convert the LPC coefficients into LSF coefficients. This is because, LSF coefficients are less prone to be affected by noise than LPC coefficients. Once we have the LSF and residuals, we train the CycleGAN for LSF and residual.

2.8.2 Observation

- The outputs in this experiment are extremely noisy and do not retain the linguistic information.
- CycleGAN is unable to effectively convert one residual coefficient to the other. This is owing to the fact that the residuals have both voiced and unvoiced parts. The unvoiced parts are mostly modelled by random noise. This prevents the CycleGAN from learning the exact conversion from one speaker to the other.

CHAPTER 3

Voice Conversion using StarGAN

StarGAN(Kameoka *et al.* (2018)) is a many to many, non parallel voice conversion model. This model is a modified version of CycleGAN so as to allow for many to many conversion unlike CycleGAN which is one to one. Similar to CycleGAN, this model doesn't require transcriptions in order to convert one voice to the other which is important since the availability of good quality transcribed data is limited to few languages.

3.1 Architecture

Unlike CycleGAN, StarGAN has a single generator, discriminator and a domain classifier in order to convert from any of the source speakers to the target speakers. Hence, we require an embedding in order to identify a given source or target speaker. The embedding could be as simple as one hot encoding or it could be the output of any speaker embedding model. As can be seen in Fig.3.1, StarGAN as well employs GateCNN. Let us denote the generator by G, discriminator by D and domain classifier by C. Akin to CycleGAN, StarGAN uses a linear transformation for the conversion of pitch from source speaker to target speaker.

3.1.1 Generator

The Generator can be seen in Fig.3.1. It consists of GatedCNN layers which take care of the sequential modelling. Unlike CycleGAN, we do not have residual layers. We have upsampling and downsampling layers which consist of convolution and de-convolution layers respectively. In the upsampling layers, the target attribute is provided so as to enable the conversion to that specific speaker.

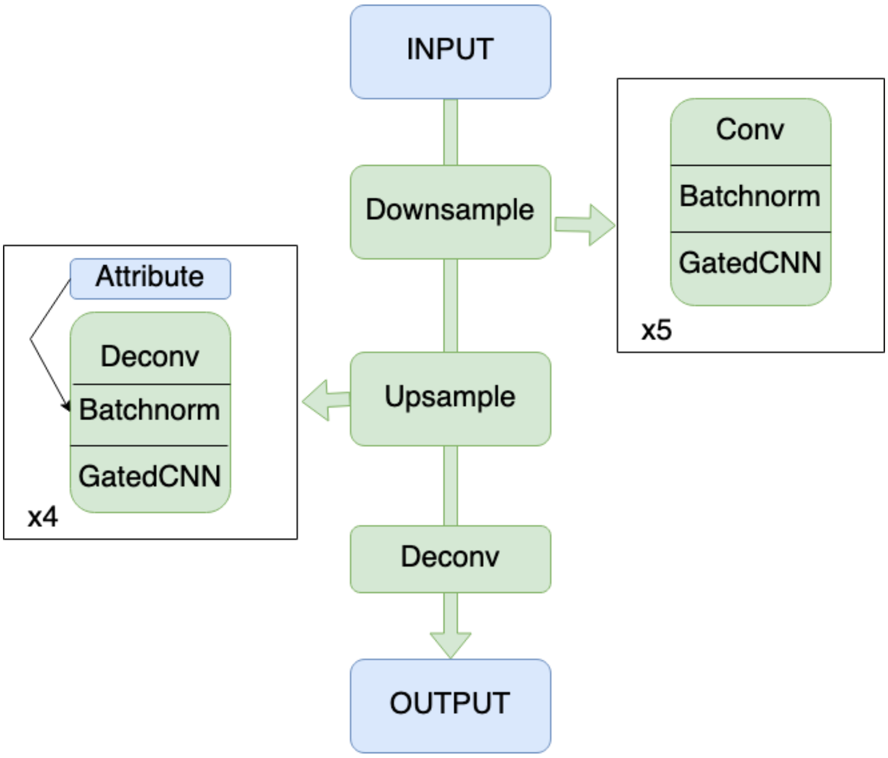

Figure 3.1: StarGAN generator architecture

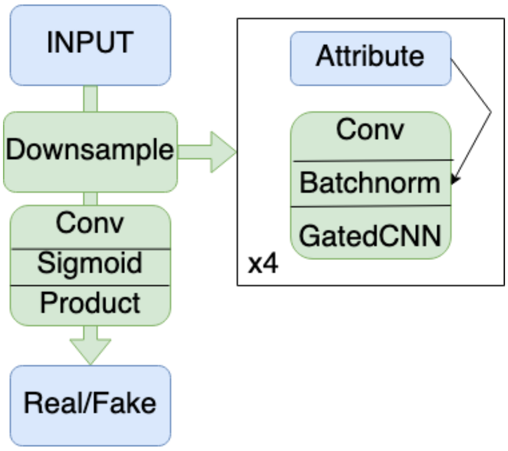

Figure 3.2: StarGAN discriminator architecture

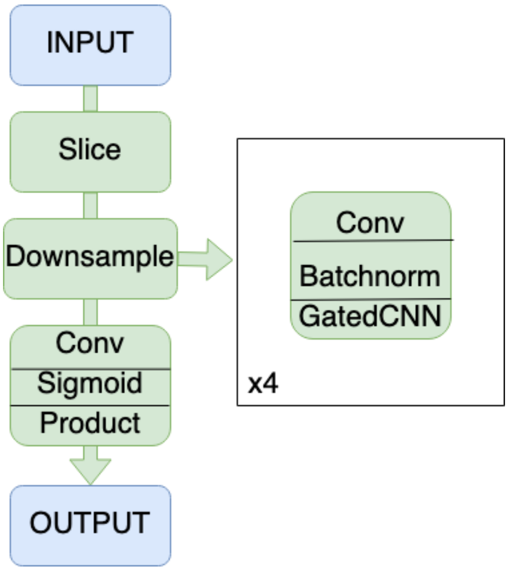

Figure 3.3: StarGAN domain classifier architecture

3.1.2 Discriminator

The discriminator takes the output of the generator or the actual speaker utterance as input and tries to classify them apart. As can be seen in Fig.3.2, it is composed of several downsampling modules with convolution and GatedCNN layers. The output of the discriminator is a single bit. The attribute of the speaker whom the utterance is supposed to belong to is provided. This enables the discriminator to model multiple such distributions for multiple speakers.

3.1.3 Domain Classifier

The domain classifier takes an input and predicts the attribute which the utterance belongs to i.e. the speaker of the utterance. In Fig.3.3, we can see that the architecture is composed of several downsampling modules composed of convolution and GatedCNN layers.

3.1.4 Loss Function

The Loss functions in the case of StarGAN is as follows:

$$L_G = L^G_{adv}(G) + \lambda_{cyc} L^G_{cyc}(G) + \lambda_{cls} L^G_{cls}(G) + \lambda_{id} L^G_{id}(G)$$

$$L_D = L_{adv}^D(D)$$

$$L_{cls}(C) = L_{cls}^C(C)$$

In the equations above, L_G, L_D, L_{cls} refer to the generator loss, discriminator loss and domain classifier loss respectively.

3.1.5 Cyclo Consistent Loss

This is akin to the loss in CycleGAN. This loss encourages the model to preserve the linguistic information. The expression for the same is as given below:

$$L_{cyc}(G) = E_{c' \sim p(c), x \sim p(x|x'), c \sim p(c)}[||G(G(x,c), c^{'}) - x||_{\rho}]$$

In the expression above, c, $c^{'}$ are the attributes of the target speaker.

3.1.6 Domain Classifier Loss

The domain classifier loss for Generator and classifier are as below:

$$L_{cls}^C(C) = -E_{c \sim p(c), y \sim p(y|c)}[\log p_C(c|y)]$$

$$L_{cls}^G(G) = -E_{c \sim p(c), x \sim p(x)}[\log p_C(c|G(x,c))]$$

From the equations above, we can see that the domain classifier loss for classifier is trying to minimize the loss when it predicts the correct attribute for a given utterance, whereas in the case of generator, it incentivises the generator to produce outputs similar to the given attribute.

3.2 Implementation Details

All the training utterances were samples at 16kHz and normalized. WORLD package is used to extract the fundamental frequency, spectral features and aperiodicities. For all the experiments, adam optimizer was used with β_1 and β_2 with values 0.5 and 0.999 respectively. The number of dimensions of MCEP features

	HinM/TamM-HinM	TamM/HinM-TamM
MCD	7.88	7.95

Table 3.1: MCD scores for HinM-TamF for 4 speakers StarGAN.

	HinM/TamF-HinM	TamF/HinM-TamF
MCD	8.13	10.02

Table 3.2: MCD scores for HinM-TamF for 4 speakers StarGAN.

used is 36. The attribute 'c' used is a one hot encoding vector. The hyperparameters λ_{cyc} and λ_{cls} are set to 10 and 10 respectively. The learning rate decays linearly after the first $2e^5$ iterations.

3.3 Experiments on Indian English

In the following experiments, we will train the StarGAN on varying number of speakers on Indian English.

3.3.1 4 speakers

The speakers which we will use include male and female native Hindi and Tamil speakers. We will include the results for one inter-gender pair i.e. HinM-TamF and one intra-gender pair i.e. HinM-TamM which will give us an idea of the general trend. The MCD values for HinM-TamM are present in Tab.3.1 and HinM-TamF are present in Tab.3.2. The cosine similarity scores for HinM-TamM and HinM-TamF are present in Tab.3.3 and Tab.3.4 respectively.

3.3.2 10 speakers

The speakers used include male and female native Hindi, Tamil, Telugu, Kanada and Gujarathi speakers. Similar to the 4 speaker case, we will include the re-

Speakers	Hin M	Tam M
Tam M - Hin M	0.63	0.5
Hin M - Tam M	0.59	0.6

Table 3.3: Cosine Similarity Scores for HinM-TamM for 4 speakers StarGAN.

Speakers	Hin M	Tam F
Tam F - Hin M	0.75	0.05
Hin M - Tam F	0.08	0.76

Table 3.4: Cosine Similarity Scores for HinM-TamF for 4 speakers StarGAN.

	HinM/TamM-HinM	TamM/HinM-TamM
MCD	10.24	10.11

Table 3.5: MCD scores for HinM-TamF for 10 speakers StarGAN.

sults for HinM-TamF and HinM-TamM in order to compare the performance with CycleGAN. The MCD values for HinM-TamM are present in Tab.3.5 and HinM-TamF are present in Tab.3.6. The cosine similarity scores for HinM-TamM and HinM-TamF are present in Tab.3.7 and Tab.3.8 respectively.

3.3.3 Observations

- We see that when the number of speakers are increased from 4 to 10, the MCD values increase considerably. This points to the fact that StarGAN though theoretically is a many to many voice conversion model, is unable to learn the mappings for extremely large number of speakers.
- When compared to CycleGAN, we see that the cosine similarity scores are better in the case of HinM-TamF.
- However, similar to CycleGAN, StarGAN too is unable to perform well in the case of same gender speakers such as in HinM-TamM case. This could be owing to the averaging effect which is amplified in the intra-gender case since the voice characteristics are much similar than the inter-gender case.

3.4 Experiment on cascaded Histogram Equalization

Similar to the CycleGAN case, we will perform Histogram Equalization after we obtain the outputs of the StarGAN. Here, we will provide results for HinM-TamM and HinM-TamF cases. StarGAN is trained on 4 speakers i.e. male and female

	HinM/TamF-HinM	TamF/HinM-TamF
MCD	7.06	10.23

Table 3.6: MCD scores for HinM-TamF for 10 speakers StarGAN.

Speakers	Hin M	Tam M
Tam M - Hin M	0.68	0.56
Hin M - Tam M	0.6	0.62

Table 3.7: Cosine Similarity Scores for HinM-TamM for 10 speakers StarGAN.

Speakers	Hin M	Tam F
Tam F - Hin M	0.73	0.1
Hin M - Tam F	0.09	0.72

Table 3.8: Cosine Similarity Scores for HinM-TamF for 10 speakers StarGAN.

native Hindi and Tamil speakers on Indian English.

3.4.1 HinM-TamM

The MCD scores for HinM-TamM can be seen in Tab.3.9. Cosine similarity scores of x-vectors can be seen in Tab.3.10.

3.4.2 HinM-TamF

The MCD scores for HinM-TamF can be seen in Tab.3.11. Cosine similarity scores of x-vectors can be seen in Tab.3.12.

3.4.3 Observations

- Akin to the case in CycleGAN, we see that there is a slight reduction in the MCD values as compared to vanilla StarGAN. However, we do not see any improvement in the cosine similarity scores.
- This shows that Histogram Equalization is not effective in terms of improving the speaker similarity in CycleGAN or StarGAN.

	HinM/TamM-HinM	TamM/HinM-TamM
MCD	7.72	7.33

Table 3.9: MCD scores for HinM-TamM Histogram Cascade experiment using StarGAN.

Speakers	Hin M	Tam M
Tam M - Hin M	0.7	0.545
Hin M - Tam M	0.48	0.59

Table 3.10: Cosine Similarity Scores for HinM-TamM Histogram Equalization cascade with StarGAN

	HinM/TamF-HinM	TamF/HinM-TamF
MCD	7.98	9.67

Table 3.11: MCD scores for HinM-TamF Histogram Cascade experiment using StarGAN.

Speakers	Hin M	Tam F
Tam F - Hin M	0.73	0.01
Hin M - Tam F	0.08	0.69

Table 3.12: Cosine Similarity Scores for HinM-TamF Histogram Equalization cascade with StarGAN

CHAPTER 4

Inferences

- In both CycleGAN and StarGAN, the model can't alter the duration of the converted utterance i.e. the converted utterance is the same duration as that of the source utterance. This is because the models process at a window level and not at an utterance level. However, we know that different speakers will have different speaking rates. Therefore, in order to improve the performance, the models must have the capability to produce variable length converted utterances.

- From the experiments on Indian languages, we can see that the models are able to perform relatively well irrespective of the language of the source and target speakers.

- In both CycleGAN and StarGAN, the use of Histogram Equalization helps in reduction of MCD scores but doesn't help in terms of improving the cosine similarity scores which is indicative of speaker similarity.

- The GAN models handle the datasets wherein the speakers are of opposite genders relatively well. However, when the two speakers belong to the same gender, the speaker similarity takes a hit. This could be owing to the fact that when the two speakers are of the same gender, both the MCEP distributions are similar to each other and the GAN model is unable to convert one to the other.

- We have observed that the converted utterances rather than having the characteristics of the target speaker alone, have the characteristics which are a mix of both the source and target speaker resulting in a different voice altogether.

- Modelling pitch using a linear transformation is not desirable since it provides a linear shift to the source pitch. However, the delta pitch is not followed as is seen in Fig.4.1. Hence, it would be a good idea to include pitch as a trainable parameter i.e. one of the inputs.

- We see that CycleGAN is able to perform crosslingual voice conversion. Although this was proposed and tested only on monolingual voices, we observe that the model is able to preserve the linguistic information and convert one voice to the other. However, the converted utterances have a large number of artefacts and are filled with noise. The CycleGAN may be able to model the crosslingual voices due to the presence of residual blocks which enable the model to learn identity extremely well. Hence, the model will be able to preserve linguistic content and change the parts which are responsible for voice characteristics.

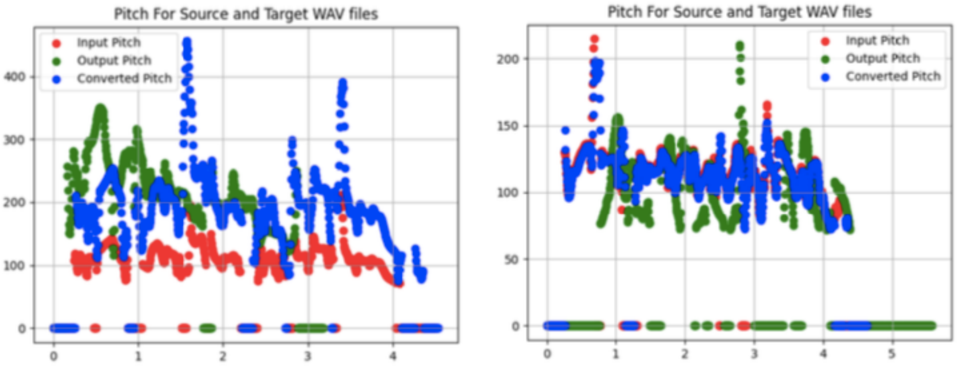

Figure 4.1: Pitch plots for the GAN models as we can see, the converted pitch and the target pitch are around the same mean as is expected from a linear transformation. However, they do not follow each other in terms of delta and delta delta features.

- In the MCEP masking experiment, we see that the performance is inferior compared to vanilla MCEP. This could be owning to the fact that including those masked features could help the model learn associations between certain linguistic features and the corresponding feature conversions, whereas in the case of the masked MCEP case, the model must learn the mapping based on the higher MCEP features alone.

- When comparing the performance of both the CycleGAN and StarGAN, we observe that the StarGAN performs better in terms of the cosine similarity scores but only marginally.

- We see that the increase in the number of speakers from 4 speakers to 10 speakers adversely affects the StarGAN performance. Although the MCD scores increase, the cosine similarity scores remain unchanged.

- Although GatedCNNs are found to work extremely well for language modelling tasks, in terms of speech they may not translate extremely well owing to their inability to capture the entire utterance prior to generation of the output i.e. the outputs are generated by looking at the windowed neighbourhood of an utterance whereas this is not the case in language modelling wherein the required window length for sentences is typically low i.e.(number of words in a sentence).

CHAPTER 5

Conclusion

Firstly, we see that GAN models are able to translate equally well to Indian languages. However, we see that GAN's suffer from inherent issues such as the inability to train well when the two distributions are extremely close in terms of distributions. Also, CycleGAN works well in the case of images owing to their fixed size whereas in the case of speech we would desire variable length outputs. In this report, we have established some baselines for GAN based voice conversion in Indian voices and have pointed out some deficiencies of GANs which are necessary to improve their performance further.

REFERENCES

1. **Baby, A.**, **A. L. Thomas**, **N. Nishanthi**, **T. Consortium**, *et al.*, Resources for indian languages. *In Proceedings of Text, Speech and Dialogue*. 2016.

2. **Dauphin, Y. N.**, **A. Fan**, **M. Auli**, and **D. Grangier**, Language modeling with gated convolutional networks. *In Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17. JMLR.org, 2017*a*.

3. **Dauphin, Y. N.**, **A. Fan**, **M. Auli**, and **D. Grangier**, Language modeling with gated convolutional networks. *In* **D. Precup** and **Y. W. Teh** (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*. PMLR, 2017*b*. URL `https://proceedings.mlr.press/v70/dauphin17a.html`.

4. **Dines, J.**, **H. Liang**, **L. Saheer**, **M. Gibson**, **W. Byrne**, **K. Oura**, **K. Tokuda**, **J. Yamagishi**, **S. King**, **M. Wester**, **T. Hirsimäki**, **R. Karhila**, and **M. Kurimo** (2013). Personalising speech-to-speech translation: Unsupervised cross-lingual speaker adaptation for hmm-based speech synthesis. *Computer Speech Language*, **27**(2), 420–437. ISSN 0885-2308. URL `https://www.sciencedirect.com/science/article/pii/S0885230811000441`. Special Issue on Speech-speech translation.

5. **Goodfellow, I.**, **J. Pouget-Abadie**, **M. Mirza**, **B. Xu**, **D. Warde-Farley**, **S. Ozair**, **A. Courville**, and **Y. Bengio**, Generative adversarial nets. *In* **Z. Ghahramani**, **M. Welling**, **C. Cortes**, **N. Lawrence**, and **K. Weinberger** (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL `https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf`.

6. **Hochreiter, S.** and **J. Schmidhuber** (1997). Long short-term memory. *Neural Computation*, **9**(8), 1735–1780.

7. **Hwang, H.**, **Y. Tsao**, **H. Wang**, **Y. Wang**, and **S. Chen**, Alleviating the over-smoothing problem in gmm-based voice conversion with discriminative training. *In* **F. Bimbot**, **C. Cerisara**, **C. Fougeron**, **G. Gravier**, **L. Lamel**, **F. Pellegrino**, and **P. Perrier** (eds.), *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*. ISCA, 2013. URL `http://www.isca-speech.org/archive/interspeech_2013/i13_3062.html`.

8. **Kabal, P.** and **R. Ramachandran** (1986). The computation of line spectral frequencies using chebyshev polynomials. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **34**(6), 1419–1426.

9. **Kameoka, H.**, **T. Kaneko**, **K. Tanaka**, and **N. Hojo** (2018). Stargan-vc: non-parallel many-to-many voice conversion using star generative adversarial networks. *2018 IEEE Spoken Language Technology Workshop (SLT)*, 266–273.

10. **Kaneko, T.** and **H. Kameoka**, Cyclegan-vc: Non-parallel voice conversion using cycle-consistent adversarial networks. *In 2018 26th European Signal Processing Conference (EUSIPCO)*. 2018.

11. **Kubichek, R.**, Mel-cepstral distance measure for objective speech quality assessment. *In Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, volume 1. 1993.

12. **Medsker, L. R.** and **L. Jain** (2001). Recurrent neural networks. *Design and Applications*, **5**, 64–67.

13. **Murthy, H.**, **R. Mohan**, **S. Srivastava**, and **A. P. Karaiyan** (2020). A hybrid hmm-waveglow based text-to-speech synthesizer using histogram equalization for low resource indian languages.

14. **O'Shaughnessy, D.** (1988). Linear predictive coding. *IEEE Potentials*, **7**(1), 29–32.

15. **Sisman, B.**, **J. Yamagishi**, **S. King**, and **H. Li** (2021). An overview of voice conversion and its challenges: From statistical modeling to deep learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, **29**, 132–157.

16. **Snyder, D.**, **D. Garcia-Romero**, **G. Sell**, **D. Povey**, and **S. Khudanpur**, X-vectors: Robust dnn embeddings for speaker recognition. *In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018.

17. **Sun, L.**, **K. Li**, **H. Wang**, **S. Kang**, and **H. Meng**, Phonetic posteriorgrams for many-to-one voice conversion without parallel data training. *In 2016 IEEE International Conference on Multimedia and Expo (ICME)*. 2016.

18. **van der Maaten, L.** and **G. Hinton** (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, **9**(86), 2579–2605. URL `http://jmlr.org/papers/v9/vandermaaten08a.html`.

19. **Vergin, R.**, **D. O'Shaughnessy**, and **A. Farhat** (1999). Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition. *IEEE Transactions on Speech and Audio Processing*, **7**(5), 525–532.

20. **Yegnanarayana, B.**, **J. M. Naik**, and **D. G. Childers**, Voice simulation: Factors affecting quality and naturalness. *In ACL*. 1984.

21. **Zhihong, W.** and **X. Xiaohong**, Study on histogram equalization. *In 2011 2nd International Symposium on Intelligence Information Processing and Trusted Computing*. 2011.

22. **Zhou, Y.**, **X. Tian**, **H. Xu**, **R. K. Das**, and **H. Li**, Cross-lingual voice conversion with bilingual phonetic posteriorgram and average modeling. *In ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019.

23. **Zhu, J.-Y.**, **T. Park**, **P. Isola**, and **A. A. Efros**, Unpaired image-to-image translation using cycle-consistent adversarial networks. *In 2017 IEEE International Conference on Computer Vision (ICCV)*. 2017.