MULTI-VOICE CLONING

Pankaj Kumar Jatav –G01338769
Prajwal Arvind Desai – G01326781
December 11, 2021

Abstract

Voice cloning is a technique to personalize speech interface. This project involves a neural voice cloning system that learns to synthesize a person’s voice from only few audio clips. There were two main approaches with this speaker adaptation and speaker encoding. In speaker adaptation fine tuning will be done on multi-speaker generative model. Speaker encoding involves training a separate model to directly generate a new speaker embedding. Both approaches can generate good results, even with few cloning audios. This project mainly has three trained components, speaker encoder network, which is trained to verify the speaker by using a separate dataset of noisy speech without the transcripts from thousands of speakers through which it generates a fixed-dimensional embedding vector from only few seconds of reference speech from a target speaker. Next is a sequence-to-sequence synthesis network, which is based on Tacotron 2 which generates a mel spectrogram from text, conditioned on the speaker embedding, and an auto-regressive WaveNet-based vocoder network that converts the mel spectrogram into time domain waveform samples. This method allows to generate speech audio similar to the voice of different target speakers, even if they were not observed during the training phase.

1 Introduction

Voice cloning is the process of generating natural speech using text. Nowadays there are several text-to-speech systems which can generate better results in terms of synthesis of natural voices which is very close to human voice. However, many of these systems learn to synthesize text using only a single voice. The main goal of this project is to build a TTS system which can generate natural speech for multiple speakers in a data efficient manner, which doesn’t require many audio samples. The activity that allows the creation of this type of models is called Voice Cloning and has many applications, such as restoring the ability to communicate naturally to users who have lost their voice or customizing digital assistants such as Alexa and Siri.

To synthesize natural speech requires training on a large number of high-quality speech-transcript audio pairs, and to support multiple speakers usually uses lot of training data per speaker. It is also not possible to record high quality data for many speakers. Our approach is to separate speaker modeling from speech synthesis by separately training a speaker-discriminative embedding network that captures speaker characteristics and training a high-quality TTS model on a smaller dataset based on what it has learned by the first network. Separating the networks will let them to be trained on independent data, which reduces the need to obtain high quality multi-speaker training data. We train the speaker embedding network on a speaker verification task to figure out if two different utterances were spoken by the same speaker. Along with the subsequent TTS model, this network is trained on untranscribed speech containing reverberation and background noise from many speakers.

2 Problem Statement

As there are most of the neural network-based text to speech synthesis are based on the single speaker. Also, they are not able to generate the speech for unseen data. Our purpose of these project to build a system which can train multiple speakers and generate voice for unseen text in training set.

We must build the system on the paper publish on google on 2nd Jan 2019 on Transfer Learning from Speaker Verification to Multi-speaker Text-To-Speech Synthesis.

Consider a dataset with n number of speakers, each of whom has multiple time-domain utterances. The mel spectrogram was chosen as a feature vector from the speaker's utterance. The speaker encoder's job is to generate meaningful embedding vectors that characterize the speakers' voices. It calculates the embedding vector for the given utterance.

3 Literature Review

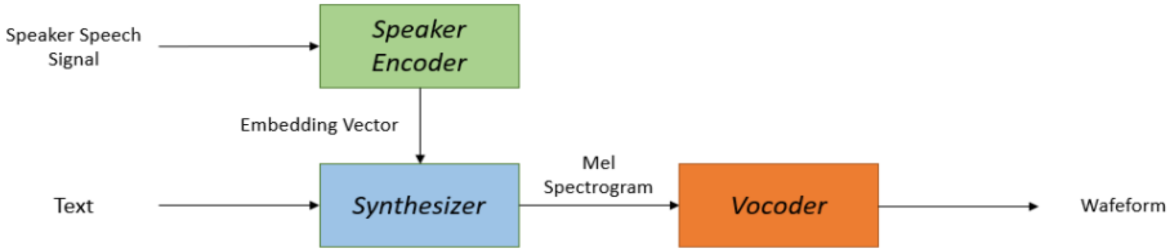
From the 12th century, people tried to build a machine to clone human voices. The existence of “Brazen Heads” includes Pope Silverster II (d. 1003 AD), Albertus Magnus (1998-1280), and Roger Bacon (1214-1294). In the 2nd half of the 18th century, The Austro-Hungarian Author and Inventor Walfgand von Kemplen build the speaking machine (Manually operated synthesizer) to produce some short sentences using a series of bellow, springs, bagpipes, and resonance boxes. In the first half of the 20th century, Bell Labs developed the first vocoder. In the mid of the 20th century, Dr. Franklin S. Cooper and Haskins Laboratories built the Pattern playback using different versions of hardware devices.

In the latter half of the 20th century, the first computer-based speech synthesis system was built by Noriko Umeda et al in Japan. The early computer-based speech synthesis methods include articulatory synthesis which simulates the human behavior of speaking using lips, tongue, moving vocal tract, and glottis, formant synthesis which produces speech based on a set of rules that control a simplified source-filter model, and these rules are usually developed by linguists to mimic the formant structure and other spectral properties of speech as closely as possible, and concatenative synthesis which concatenation of speech pieces from the previously-stored voice in database.

Later, the hidden Markow model (HMM) based methods which are also known as STATISTICAL PARAMETRIC SPEECH SYNTHESIS is proposed, which predicts the spectrum, frequency, and duration for speech synthesis. From the 2010s, neural network-based speech synthesis becomes more popular as the computation speed of computers increase gradually. The neural network-based system adopts (deep) neural networks as the result WaveNet is proposed which directly generate the waveform from the linguist features.

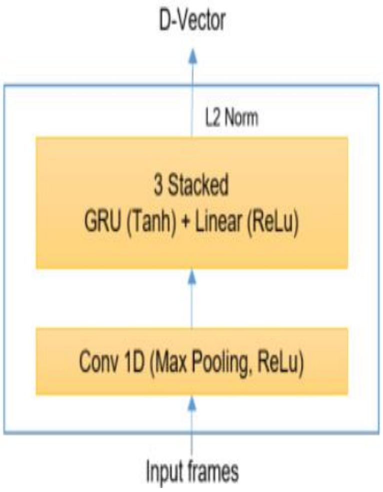
4 Methods and Techniques

The system consists of three independent of trained neural networks components, illustrated in the below figure.



Speaker encoding:

The speaker encoder is used to produce an embedding vector (d vector) that transformed the reference speech signal to desired target speaker. Furthermore, the model should capture the different characteristics of the different speakers using short speech signals, regardless of their phonetic content and background noise. And these can be met using a neural network model on a text-discriminative speaking verification model that is trying to minimize the GE2E loss So that embedding of utterances from the same speaker has high similarity and low distance and low similarity and high distance from other users. The network maps a sequence of mel spectrogram frames to a fixed-dimensional embedding vector, known as d-vector. The input frame (Mel spectrograms) is fed to a Conv1D layer of 256 units. And the output of it fed to 3 stack GRU (512 units) and Linear Projection of 256 dimensions which can be seen in the below figure. And we found that this architecture is best for our problem.



Synthesizer and Vocoder:

The synthesizer component is trained on pair of text sequences and audio Mel spectrogram sequences. To do this we used the Tacotron 2 architecture to support multiple speakers.

To invert synthesized Mel spectrogram emitted by the synthesis network into the time-domain waveform we used WaveNet as a vocoder which is composed of 30 dilated convolution layers. The network does not depend on the speaker encoding. The mel-spectrogram uses the synthesizer network which captures all the relevant details needed for high-quality voice synthesis of a different user. And this allows multispeaker vocoder based on training on multiple speaker's data.

5 Discussion and Results

5.1 System Configuration

We have used the following system and versions of cuda to train our model. All the result and time duration will be based on our below system configuration.

Component	Version/ Size
OS	Arch Linux
Kernal	5.10.81-1-lts
CPU	AMD Ryzen 7 5800X
GPU	NVIDIA GeForce RTX 3070
CUDA Version	11.5
RAM	32GB

Table: System info

5.2 Datasets

For this project, we have used the free available audio of book reading on LibriSpech. The data is present in different size based on hours of recording and number of speakers. The recordings are carefully segmented and aligned. Most of the recording does not contain any noise or background sound. Thus, the data the clean and require minimal or no audio prepossessing. The data set also contains the transcript for each audio.

Vassil Panayotov worked with Daniel Povey to create LibriSpeech, a corpus of around 1000 hours of 16kHz read English speech. The information comes from the LibriVox project's read audiobooks, which have been carefully separated and aligned.

Dataset	Size	Duration	Utterance	Total Speakers	Male	Female
train-clean-100	6.3GB	100 hours	27950	251	126	125
train-clean-360	23 GB	360 hours	101878	921	482	439
train-other-500	30 GB	500 hours	95708	1166	602	564
Data used for Training						
train-clean-100	6.3GB	100 hours	27950	251	126	125
train-other-500	30 GB	500 hours	95708	1166	602	564

Table: Data subsets in LibriSpeech

5.3 Evaluation Metrics

We used the Speaker Verification Equal Error Rate (SV-EER) to evaluate all the speaker encoding. IT was estimated by pairing each test utterance with each speaker from the dataset. We have also used the average loss matrix to validate our model.

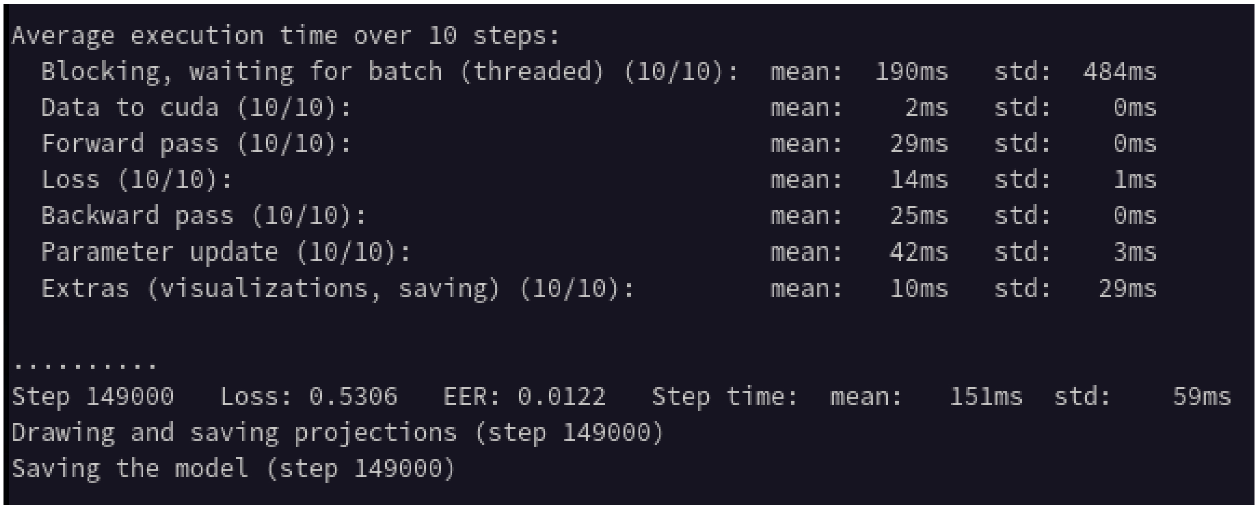
We used the single speaker matrix to compute estimate the time of model training and see how the model is performing.

We used visdom package to print our matrix for SV-ERR and average loss matrix during the training time. This also allows us to compare models with different parameters.

For Speaker Synthesizer we plot Mel-spectrogram and attention after each fixed value of steps depending upon the size of the train dataset

System	Training Speaker	SV-EER	Average Loss
Single Speaker	1	0.5% +- 0.2%	4.15%
Multiple Speaker	1166	0.03 +-0.002	0.32%
Multiple Speaker without encoding	1166	0.04 +- 0.04	0.55%

Table: Comparing different systems using SV-ERR and Average Loss



```
Average execution time over 10 steps:
  Blocking, waiting for batch (threaded) (10/10):  mean:  190ms   std:  484ms
  Data to cuda (10/10):                            mean:    2ms   std:    0ms
  Forward pass (10/10):                            mean:   29ms   std:    0ms
  Loss (10/10):                                    mean:   14ms   std:    1ms
  Backward pass (10/10):                           mean:   25ms   std:    0ms
  Parameter update (10/10):                        mean:   42ms   std:    3ms
  Extras (visualizations, saving) (10/10):         mean:   10ms   std:   29ms

..........
Step 149000   Loss: 0.5306   EER: 0.0122   Step time:  mean:   151ms  std:    59ms
Drawing and saving projections (step 149000)
Saving the model (step 149000)
```

Screenshot of model training with after 10 steps to verify how the model is training

5.4 Experimental Results

We had trained the model for different parameters. We have trained the model using single speaker, multiple speakers with embedding vector and multiple speakers without embedding vector. We have found out that that multiple speakers with encoding outperformed all the different models in term of Speaker Verification Equal Error Rate (SV-EER) and average loss.

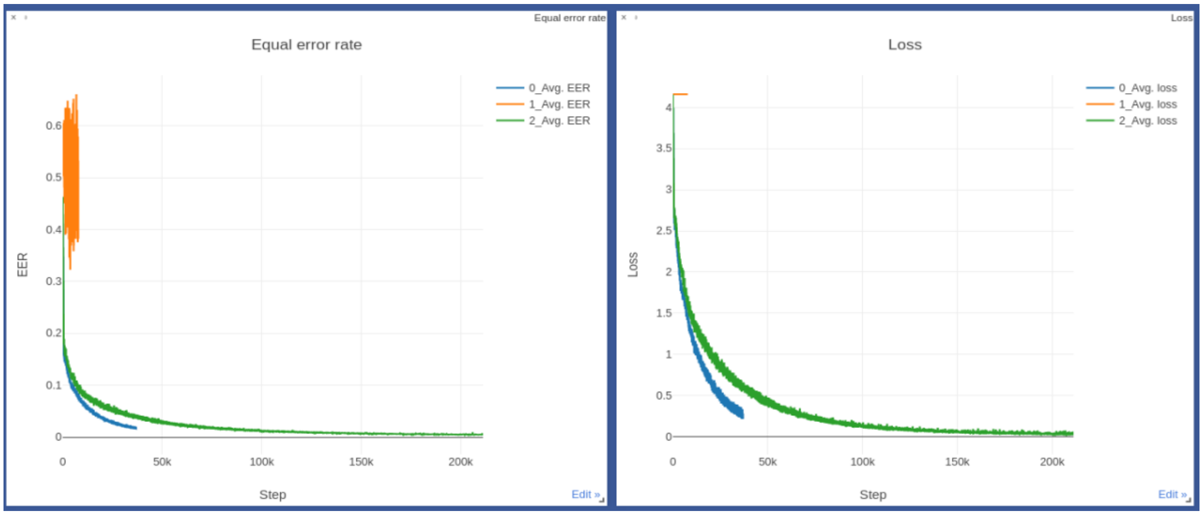


Fig: A comparison of different models

With the training time and number of steps the segregation of speakers changed, which can be observed at the below image.

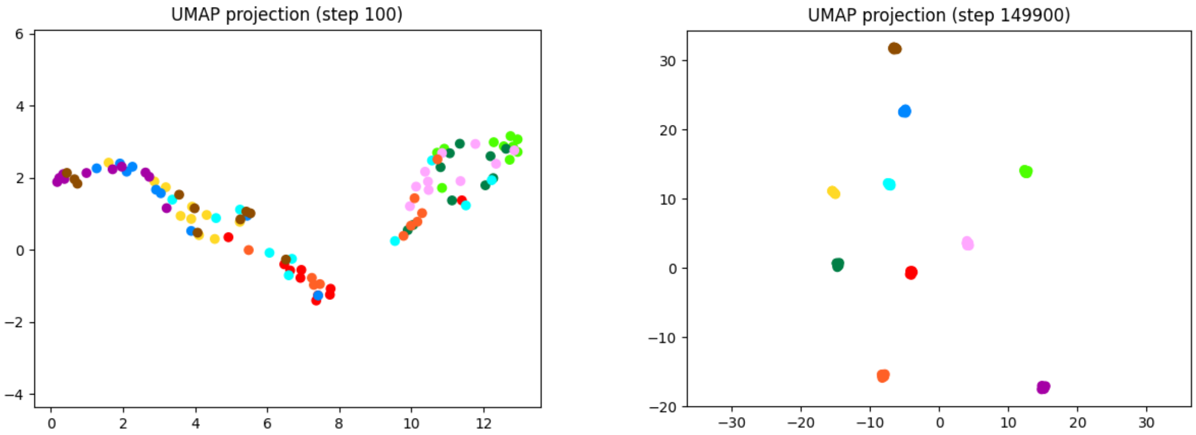
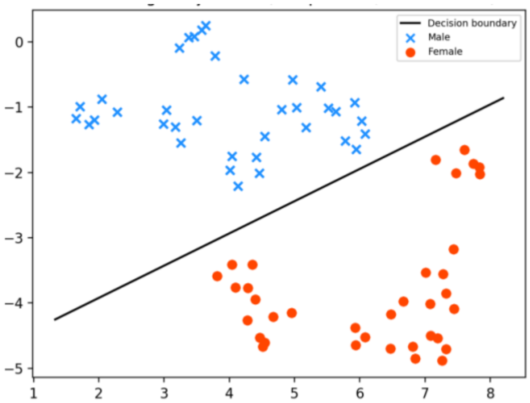


Fig: segregation of speakers over time

We are also able to successfully segregate speaker embedding of male and female from dataset utterance, which can be seen in the figure.



As we used the used the Tacotron as our voice synthesizer the Target and Predicted Mel-Spectrogram loss decrease with the training time and step and expected. Which can be seen at below figures.

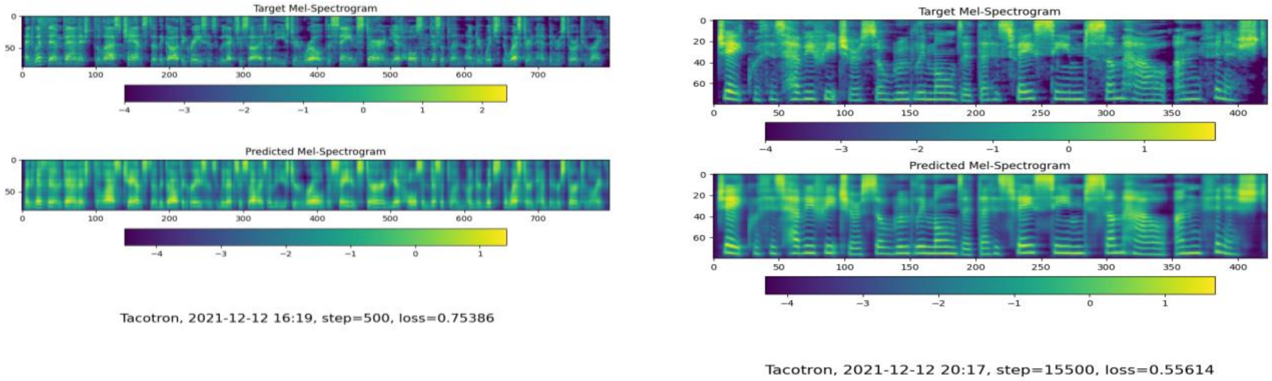


Fig: Mel Spectrogram of 500 and 15500 step

Here are some output samples from our project:

Input Audio:

1. Input Audio Link

Output Audio:

1. Output Audio Link 1
2. Output Audio Link 2
3. Output Audio Link 3
4. Output Audio Link 4 (Using Griffin-Lim Vocoder)

6 Conclusion

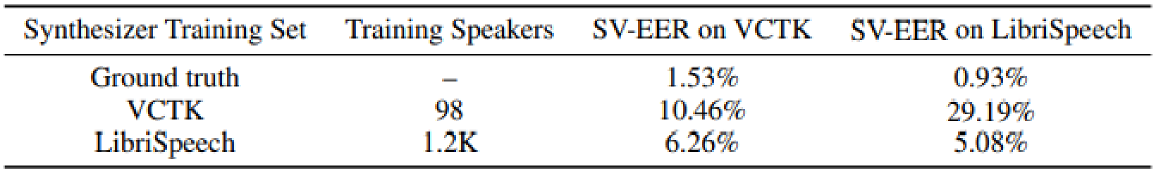
We have developed a neural network-based system for multispeaker TTS synthesis. The system combines a speaker encoder network that has been independently trained with a sequence-to-sequence TTS synthesis network and a neural vocoder based on Tacotron 2. The synthesizer can generate high-quality speech not only for speakers seen during training, but also for speakers never seen before, by using the discriminative speaker encoder's understanding. We demonstrated that the synthesized speech is reasonably similar to real speech from the target speakers, even on unseen speakers, using evaluations based on a speaker verification system as well as subjective listening tests.

In contrast to the single speaker results, the proposed model does not achieve human-level naturalness, despite the use of a WaveNet vocoder. This is due to the increased difficulty of generating speech for a wide range of speakers with significantly less data per speaker, as well as the use of datasets with lower data quality. Another drawback is the model's inability to transfer accents. This could be addressed by conditioning the synthesizer on independent speaker and accent embeddings if enough training data is available.

Finally, we show that the model can create realistic speech from fake speakers who aren't in the training set, signaling that the model has learned to use a realistic representation of speaker variation space.

In addition to the paper that we used to build this project. We have built the system with low SV-EER which can be see below.

Table 4: Speaker verification EERs of different synthesizers on unseen speakers.



Synthesizer Training Set	Training Speakers	SV-EER on VCTK	SV-EER on LibriSpeech
Ground truth	–	1.53%	0.93%
VCTK	98	10.46%	29.19%
LibriSpeech	1.2K	6.26%	5.08%

Table: Comparing different systems using SV-ERR and Average Loss from our paper

System	Training Speaker	SV-EER	Average Loss
Multiple Speaker(LibriSpeech)	1166	0.03 +-0.002	0.32%

Table: Comparing different systems using SV-ERR and Average Loss from our project

6.1 Directions for Future Work

Below are the following points for our future work on this project:

- The model performed much better when the data seemed to be clean and free of noise, but it did not perform well when the data was noisy. As a result, we want to improve our model's performance when dealing with noisy data.
- Explore and design the same project using CNN and Deep learning as DeepVoice 3 can clone the voice using 3.5 seconds audio sample.
- Explore the speakers with multiple language dataset and evaluate the model performance on multi-language multi-speakers' dataset.

References

[1] *Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis*: https://arxiv.org/pdf/1806.04558.pdf

[2] *Anthropomorphic Talking Robot Waseda Talker Series:* http://www.takanishi.mech.waseda.ac.jp/top/research/voice/index.htm

[3] *Articulatory synthesis:* https://en.wikipedia.org/wiki/Articulatory_synthesis

[4] *Formant*: https://en.wikipedia.org/wiki/Formant

[5] *Formant Synthesis Models: https://ccrma.stanford.edu/~jos/pasp/Formant_Synthesis_Models.html*

[6] *Concatenative synthesis: https://en.wikipedia.org/wiki/Concatenative_synthesis*

[7] *Hidden Markov model: https://en.wikipedia.org/wiki/Hidden_Markov_model*

[8] *Statistical Parametric Speech Synthesis: https://www.cs.cmu.edu/~awb/papers/icassp2007/0401229.pdf*

[9] *Statistical Parametric Speech Synthesis Using Deep Neural Networks: https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/40837.pdf*

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