Juni Khyat
(UGC Care Group I Listed Journal)
ISSN: 2278-4632
Vol-13 Issue-02 July 2023

REAL TIME VOICE CLONING

T. Revanthi Chandra, B.S Sai Kiran, Dr. G. Sudhakar
SWARNANDHRA College of Engineering and Technology
Department of Information Technology

***Abstract*:** Through training and without retraining the template, a new research has developed a three-level pipeline that can clone an unseen voice from a few seconds of reference speech. Findings that sound very natural were found by the researchers. The goal is to make a public copy of this model and make it open source. The goal of modifying the framework for a new vocoder version is to enable real-time operation. In order to carry out real-time voice cloning, it is necessary to deepen one's understanding of the device by three levels. When it comes to localizing speech to text, the latest advancements in Deep Study have shown remarkable outcomes. Given this, it's very uncommon for a single speaker to make up a deep neural community, all using a professionally recorded audio corpus. Retraining this version of the model is an expensive ordeal that necessitates gathering a fresh dataset. There is currently just one public implementation of this framework, which is the end product of Google's 2018 paper. Even with just five seconds of speech, the system may want to record a virtual version of the speaker's voice. Having removed the written content, you are free to utilize any voice recovered from this method for text-to-speech conversions. Excited for the three phases of release, whether it's our own implementation or one of the open-source implementation plans.The goal is to apply a well-informed model of the relevant pipeline for data preparation. After then, using massive datasets including tens of hours of audio, these models representing thousands of audio systems will need to be trained for weeks or months. Look at their advantages and disadvantages instead. First and foremost, we know that this technology is real-time, meaning that we can record and create audio in a fraction of the time it takes for the actual audio to be created. At some point during training, the framework may be able to reproduce sounds you've never heard before and perhaps create new sounds from text you've never seen before.

***Keywords*: Neural Networking, 3-stage pipeline, Text-to-Speech, Deep Learning.**

INTRODUCTION

In several branches of computational machine learning, a thorough understanding of models has been shown. Methods such as text-to-speech (TTS) synthesis, which involves transforming text into speech, are not an exception. Deep epidemics in 2016 might generate more natural speech than conventional consolidation methods. There has been a lot of work on training in a stop-to-give fashion, enhancing the performance of these low-end models, and making the sound more natural. Many instances run slower than real time on mobile CPUs, which is why the example is on GPU. Surprisingly, subjective criteria work better when evaluating the naturalness of speech. Because it sounds so much like real human speech, we may assume that "a speech that is more natural than human speech" exists. While it's true that we've technically hit our limit as a species when it comes to obtaining a TTS version on a single speaker, the point is to use a fixed version and find a way to include your new voice with less data, rather than relying on a voice clone. Tuning a TTS template that has been taught to generalize speech and then applying it to fresh audio is one such approach. It is from the speaker-encoder variant that takes the reference audio as input that the low-dimensional embedding is produced. Compared to making a unique TTS version for every speaker, this method is often quicker, more efficient, and uses less computing power. It seems that different techniques may have significantly different reference speech lengths, which can range from 30 minutes to a few seconds depending on the speaker. This reference speech duration is crucial for voice cloning. From thirty minutes to a few seconds, this is often what decides how close the cognitively produced speech is to the actual voice of the speaker, which is essential for cloning a voice among the many approaches. The degree to which the generated voice resembles the speaker's actual voice is often dictated by this feature.

ISSN: 2278-4632
Vol-13 Issue-02 July 2023

LITERATURESURVEY

Paper I: Real Time Voice Cloning

In just a few seconds during practice, you may duplicate reference audio's invisible audio using a three-step pipeline—all without retraining the template. A highly realistic-sounding audio output was provided by the researchers. Making a copy of this model and making it accessible to the public is the goal of this effort. In order for the framework to function in real time, the new vocoder model modifies it. Creating a three-stage deep learning system capable of real-time voice cloning is the aim.

Paper II: Neural Voice Cloning With a Few Samples

Among the many features that make customized speech interfaces so appealing is voice cloning. Research has shown that neural network-based voice synthesis can generate high-quality speech from a huge pool of speakers. A neural voice cloning method that requires less audio samples as input is introduced in this article. Speaker encoding and speaker adaptability are two methods that we examine. The process of speaker adaption involves honing the produced multi-speaker model with the help of several clone samples. In order to encode speakers, one must first train a different model to infer a new speaker embedding from the duplicated audio, which is then used in a model that generates audio with multiple speakers.

Even when just a little amount of cloned audio is used, both methods produce convincingly authentic speech that is similar to the original. For deployments with limited resources, speaker adaption is a good option since it improves naturalness and similarity while using much less memory and cloning time than the speaker encoding method.

Paper III: WAVENET: A Generative Model For Raw Audio

One deep neural network that can generate unprocessed audio waveforms is WaveNet, which is introduced in this article. The expected distribution of every audio sample is dependent on every sample that came before it; the model is entirely autoregressive and based on probability. Even so, it demonstrates that training data with tens of thousands of samples per second of audio can be done effectively.It offers state-of-the-art performance in text-to-speech applications and, according to human listeners, sounds much more realistic than the top parametric and concatenated systems for English and Mandarin. It is possible to accurately record the characteristics of several speakers with a single WaveNet and switch between them simply changing the speaker ID. We discover that educated to model music, it creates original and often very accurate musical compositions. As a discriminative model, it has shown encouraging performance in phoneme identification tasks.

Paper IV: Efficient Neural Audio Synthesis:

Results for data distribution estimates and high-quality sample creation in the audio, picture, and text domains are provided by the Sequential Model that are state-of-the-art. Nevertheless, the challenge of efficiently sampling this class of models is still unsolved. We will go over a variety of standard methods for decreasing sample time without sacrificing output quality, with an emphasis on text-to-speech synthesis. We will start with WaveRNN, a state-of-the-art recurrent neural network that is one layer deep and uses a double softmax layer, to compare it to the state-of-the-art WaveNet model. Because the network is so small, graphics processing units (GPUs) can produce 24 kHz 16-bit audio four times quicker than in real time. After that, shrink the WaveRNN's weights by using a weight pruning method. At least for certain parameter values, big sparse networks outperform tiny dense ones. For sparsity levels greater than 96%, this correlation holds as well. Sparse WaveRNN's small weight count enables real-time sampling of high-fidelity audio on mobile CPUs. At last, we propose a sub-scaling-based novel generation technique. This allows for the simultaneous generation of many samples by reducing a lengthy sequence to a stack of shorter sequences. Subscale WaveRNN offers an orthogonal approach to improve sample efficiency and generates 16 high-quality samples in a single phase.

Paper V: Natural TTS synthesis by conditioning Wavenet onMel spectrogram predictions:

This study showcases the Tacotron 2, an architecture for neural networks that can synthesize voice from text. Character embeddings are mapped to Melscale spectrograms using an iterative inter-sequence functional prediction network, and time domain waveforms are synthesized from these spectrograms by a modified WaveNet model, which functions as a vocoder in the system. The Mean Opinion Score (MOS) for our model is 4.53. Compared to MOS 4.58, this is a good option for professionally produced audio. We assess the effect of

ISSN: 2278-4632
Vol-13 Issue-02 July 2023

feeding the Mel spectrogram into WaveNet instead of the voice, duration, and F0 functions, and we provide an ablation analysis of the system's critical components to support the design choice. There is room for significant simplification in the WaveNet design, as shown by a small intermediate acoustic representation.

Paper VI: VOICE CLONING: A Multi-Speaker Text-To-Speech Synthesis:

Many subfields of machine learning are starting to use deep learning models. One such example is text-to-speech, which involves generating synthetic voice from text. This is why it is common practice to train deep neural networks using hours of audio recordings made by a single speaker. It is time-consuming and expensive to try to create a new speaker voice from scratch by acquiring a fresh dataset and retraining the model. Because of this, most TTS versions are designed with only one speaker. To get over these restrictions, the suggested method is to build a system that can simulate an acoustic area with several speakers. Even if the target speakers weren't there during training, you may nevertheless create sounds that are similar to theirs.

PROPOSED SYSTEM

The basic components of the whole program can be expressed as follows.

The important elements are:

- Feature Extractor: -The role of the feature extractor is to provide data that better shows how the voice produced by the model sounds.
- Acoustic Model:- The relationship between the calculated features and the output acoustic features for the input text is learned by a statistical generative model called the acoustic model.
- Vocoder: A system that can reconstruct audio waveforms from the acoustic features generated by an acoustic model. Derivate the audio waveform from the spectrogram generated by the synthesizer.
- Speaker Encoder: Derives embedding from short utterances of a single speaker. Embedding is a meaningful expression of the speaker's voice, where similar voices approach each other in potential space.
- Synthesizer: The embedded speaker creates a spectogram from the text. This model is the popular Tacotron 2, which does not uses the WaveNet.

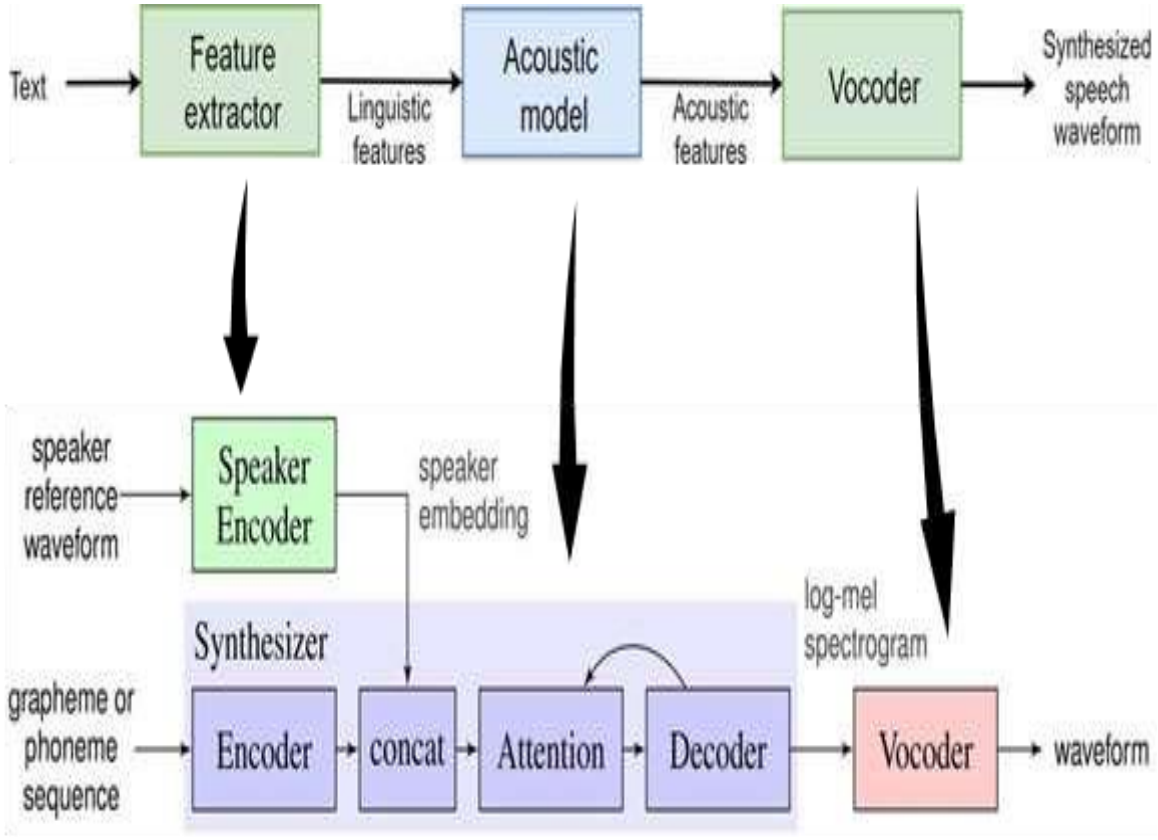


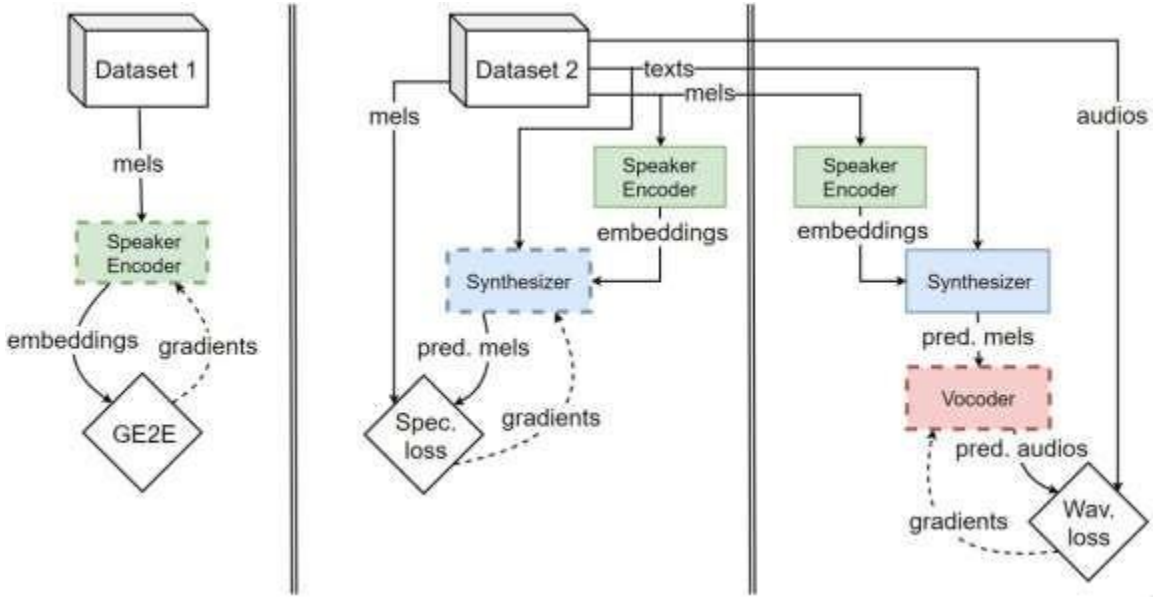
Figure 1: System Block Diagram

ISSN: 2278-4632
Vol-13 Issue-02 July 2023

FLOW CHART

The preceding figure provides a more detailed overview of the procedures that will be used to carry out our project. Using the input audio, the encoder generates voice embeddings that mimic the speaker's natural speech pattern. The Synthesizer takes text input and uses machine learning methods to construct graphemes or phoneme sequences. Without retraining the system, the vocoder uses the Mel-Spectrogram generated by the encoder and synthesizer to provide the final cloned voice output in the most aspirated voice.

Figure 2: The System Flowchart



IMPLEMENTATION

V.I Implementation

Installing prerequisites, setting up the environment for the project to work, collecting datasets, encoding and implementing encoder, synthesizer, and vocoder modules are all part of the system-wide implementation. All of the necessary code and implementation was done in Python 3, as previously stated.

V.II Installations

The mandatory installations that are required for the working of the project included
•TensorFlow GPU(1.10.0=<Version<=1.14.0)
•Umap-learn
•Visdom
•Webrtcvad
•Librosa (0.5.1=<Version)
• Sounddevice
•Unidecode
•PyTorch
•Inflect

V.III Python pip in the Python shell was used to do the installation described above. The next step, when the installation is finished, is to set up the new installation so that we may operate in an atmosphere that is suitable for our projects. The demo_cli.py file is used for this purpose. It is considered a successful installation and configuration of prerequisites when the demo cli.py file runs without any issues. Dataset used

ISSN: 2278-4632
Vol-13 Issue-02 July 2023

On the SV2TTS, researchers have uncovered two datasets that can be used to train vocoders and synthesizers. These include the aforementioned LibriSpeechClean as well as the VCTK10, which was recorded using professional equipment and include a panel of 109 native English speakers. The VCTK audio is downsampled from 48 kHz throughout the test. Compared to LibriSpeech's 16 kHz sampling, this is still better. When compared to VCTK, synths taught using LibriSpeech are more comparable to the average population, but they sacrifice linguistic realism. It was easy to get the database download link as the record was publicly available and all it took was an inquiry email to the University of Edinburgh, the record's original owner.

V.IV Implementation of Speaker Encoder

VI The Speaker Encoder is the first module that needs training. It incorporates the preprocessing, audio training, and visualization models, and manages the audio input given to the system. A LSTM3-layer with 768 hidden nodes and a projection layer with 256 units makes up the Speaker Encoder. The publications did not define a projection layer, therefore we are assuming it is just a densely networked layer with 256 outputs per long short-term memory (LSTM) that is applied repeatedly to each LSTM output. To simplify things, reduce training load, and speed up prototyping, you may utilize 256 LSTM layers directly instead of building the speaker encoder from scratch. A 40-channel log-mel spectrogram with a 25 ms window width and a 10 ms stage is the result we get here. An output vector of 256 elements is produced by the final layer, which is the L2-normalized hidden state.We have also included an easier-to-understand pre-standardization ReLU layer in our implementation, which aims to make embedding sparse.

V.VI Implementation of Synthesizer

The Google Tacotron 2 model, devoid of Wavenet, is used as the synthesizer. Tacotron is a system that predicts text-based mel spectrograms via iterative intersequences. A vector of certain characters is first added from the text string. Then, to make one encoder block longer, regular layers are added. In order to generate encoder output frames, these input frames are passed through bidirectional LSTMs. At this stage, SV2TTS modifies the existing architecture. The Tacotron encoder attaches the speaker embedding to every frame it generates. By analyzing the encoder's output frame, the attention function creates an input frame for the decoder. We don't validate the input text's pronunciation in our implementation; the characters are given without any checks. There are still a few things to do while cleaning. Transform all numerals and acronyms into full-text format; convert all letters to ASCII; standardize spaces; and decrease all characters. You are free to use punctuation marks, but they will not be included in the record.

V.VI Implementation of Vocoder

The Encoder Synthesizer Vocoder is anticipated to educate the subsequent modules in the sequence, hence, the Vocoder modules undergo training last. WaveNet is a voice changer compatible with Tacotron 2 and SV2-TTS. Based on WaveRNN but offering many user fatchord design choices, the vocoder model used is the free source PyTorch implementation 15.An acronym for "Alternative WaveRNN" describes this design. Each training phase uses the same number of segments to split the mel spectrogram and its accompanying waveform. The inputs for the design are the simulated spectrogram segments t and t-1. A well-designed one will generate identically sized waveform segments, denoted as t. While keeping the total number of mel channels constant, the upsampling network adjusts the mel spectrogram such that it fits the target waveform's duration. After converting the mel spectrogram to a waveform, models such as resnet take it as an input and use it to create features that modify the layers. Iteratively adjusting the length of the waveform segment is accomplished by using the generated vector. After that, the adjustment vector is split into four equal parts according to the channel dimensions; the first component is joined with the upsampling spectrogram and waveform segment from the prior time step. The resultant vector is subject to some modifications when a skip connection is used. Two GRU layers are placed first, followed by a high density layer.

ISSN: 2278-4632
Vol-13 Issue-02 July 2023

TESTING AND DEBUGGING

To make sure the initial model worked well before implementing it into a wider workflow, it was evaluated in Google Collab, an online machine-learning environment supplied by Google. At first, testing ran into problems with audio input difficulties and settings. Those were addressed following the appropriate problematic code parts. Below, you will find the attached screenshots of the testing process:

Figure 3: Uploading the project for testing on Google Collab

Figure 4: The output obtained on Google Collab while testing.
This signified successful testing of initial project

DEBUGGING

Environmental and installation-related problems dominated the testing and implementation phases. Installing PyTorch and Webrtcvad again, this time with less stringent requirements and less tweaks, helped a lot. We also found an issue with the toolbox that was used to launch an instance, which would then close itself instantly. After some digging, we found out that the record path was missing the root directory, which was causing this issue. I discovered the system was functioning correctly after running the necessary root directory command. The remaining mistakes I found were minor and could be resolved by reviewing the code.

RESULT

Because the project ran well, the system was able to use text-to-speech synthesis to accurately replicate a specific voice file. There is no way to put a number on how accurate the findings are since voice quality is a subjective attribute. One good strategy is to find out how people feel about the cloned audio output in comparison to a real human voice by conducting a poll and calculating the Mean Opinion Score (MOS). An analysis of MOS reveals that the cloned voice sounds quite similar to the original human voice, with the exception of a loss of naturalness and accent—two editable characteristics. In addition to providing an opportunity for the project to be developed if needed, the project's shell functionality verifies that the processes are functioning as intended. The framework

ISSN: 2278-4632
Vol-13 Issue-02 July 2023

may be easily obtained by users using a graphical user interface, eliminating the need for extensive research. Its official name is the "SV2TTS toolbox." Figure 8.2 shows the toolbox interface. It is cross-platform since it is developed in Python using the Qt4 graphical user interface.

Figure 5: The SV2TTS Tool box used for graphical interfacing of the project.

There are a lot of common language records in the toolbox, and you may add them to new records by editing existing ones. Further, the customer has the option to record the statement and make an exact copy of his speech. Automatic updates to the UMAP projection and calculation of the utterance's embedding occur upon loading. There has been no measurement, hence the mel spectrogram of the speech is shown as a comparison only (central row on the right). The square form is structurally meaningless in relation to the embedding value since the embedding is a one-dimensional vector. You can see the difference between the two embeddings graphically when you draw them.From the top right of the user interface, the client may input any text for synthesis. Just a friendly reminder that the template has been deprecated and does not support punctuation. The user may modify the produced utterance's rhythm by inserting line breaks between the separately synthesized sections. When these components are joined, they form a full spectrogram. After the spectrogram is composited, it will be shown in the interface's bottom right corner.The customer finishes by creating a segment using the vocoder to react to the synthetic spectrogram. The generation's development may be seen in the import bar. After everything is finished, UMAP is used to project the composite utterance embedding that is shown on the left side of the composite spectrogram. Customers have the option to use embedding as a guide for the future.

OpinionSet	Naturalness	Similarity	Source
VCTK	4.28 ±0.05	1.82 ± 0.08	Jiaet al 2018 paper
LibriSpeech	4.01 ± 0.06	2.77 ± 0.08	
ThisProject	4.10 ± 0.12	2.19 ± 0.20	Survey(25 people)

Table 1 : Final MOS Results obtained from a personal survey

The table above is a summary of the final results. It attempts to give the highest subjective measure that can be obtained to understand the quality of the audio produced by our project.

CONCLUSION

Without publishing the results, this research has created a foundation for real-time voice cloning. The findings are enough, and the framework does a reasonable job of simulating speech—at least up to the point where we can figure out how to employ more reference speech time—despite some strange prosody. Absolutely not. Even when this project is over, there are still methods to make certain frameworks better and apply some of the new developments in this field. Speech generation was enhanced in the aforementioned initiatives and studies by use of state-of-the-art deep learning networks. Everyone agrees that our design and toolkit is an upgraded version of the TTS prototype, but we can also say with certainty that subsequent models in the same technological field will be much better. As a result, this strategy turned out to be an effort to comprehend, use, and develop upon

ISSN: 2278-4632
Vol-13 Issue-02 July 2023

the knowledge the organization had acquired via its study. We think that stronger varieties of The day will soon come when you can purchase a voice clone.

REFERENCES

[1] The authors of the passage are Gilles Loupe and Corentin Jemine. I am writing my master's thesis on automatic multi-speaker voice cloning. The University of Liege's Faculty of Science Applications. Source URL: http://hdl.handle.net/2268.2/6801.

[2] Alexander Graves, Koray Kavukcuoglu, Andrew W. Senior, Nal Kalchbrenner, Aaron van den Oord, Heiga Zen, Sanders Dieleman, and Andrew W. Senior. Wavenet: A paradigm for generating unprocessed audio. The Citation: CORR, 2016 (abs/1609.03499), http://arxiv.org/abs/1609.03499.

[3] Aaron van den Oord, Florian Stimberg, Karen Simonyan, Seb Noury, Norman Casagrande, Nal Kalchbrenner, Erich Elsen, Edward Lockhart, Sanders Dieleman, and Koray Kavukcuoglu all contributed to this work. A neural network for efficient audio synthesis, 2018.

[4] In the following list of authors: R. J. Skerry-Ryan, Rif A. Saurous, Jonathan Shen, Ruoming Pang, Ron J. Weiss, Yuxuan Wang, Yannis Agiomyrgiannakis, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, and Yonghui Wu. Predicting mel spectra with wavenet conditioning allows for natural TTS synthesis. For the 2017 version, see http://arxiv.org/abs/1712.05884 in the journal CoRR.

[5] Gianluca Penn and Stefano Shirali-Shahreza. Mos-naturalness and the pursuit of speech that mimics human language. Pages 346–352 of the 2018 IEEE Spoken Language Technology Workshop (SLT), December 2018. article DOI: 1109/SLT.2018.8639599.

[6] Yanqi Zhou, Kainan Peng, Jonathan Raiman, John Miller, Sercan Arik, Gregory Diamos, and Andrew Gibiansky. In 2017, Deep Voice 2 released its multi-speaker neural text-to-speech system.

[7] Vincent Pollet, Luigi Di Caro, Giuseppe Ruggiero, and Enrico Zovato. Voice Cloning: A Transfer Learning-Based Method for Synthesizing Speech from Multiple Speakers, 2021. To access the article, go to https://arxiv.org/abs/2102.05630.

[8] "Real Time Voice Cloning" by Corentin James, 2019.

[9] Yanqi Zhou, Sercan O. Arik, Jitong Chen, Kainan Peng, Wei Ping, and Jitong Chen. Artificial Intelligence Voice Synthesis with a Small Sample Size, 2018. The link to the article is https://arxiv.org/abs/1802.03006.
The authors of the report are Guanglu Wan, Guoqiao Yu, Lei Xie, Shan Yang, and Jian Cong. Voice cloning from noisy samples using domain adversarial training for efficient data, 2020.