TEXT-TO-SPEECH WITH CUSTOM VOICE

Abstract

The Text to Speech with Custom Voice system has vast applicability in numerous industries, including entertainment, education, and accessibility. The proposed text-to-speech (TTS) system is capable of generating speech audio in custom voices, even those not included in the training data. The system comprises a speaker encoder network, a synthesizer network, and a WaveRNN vocoder network. Multiple speakers from a dataset of clean speech without transcripts are used to train the speaker encoder network for a speaker verification task. It creates a fixed-dimensional embedding vector using the target speaker's reference speech. Using the speaker embedding, the synthesizer network based on Tacotron2 creates a mel spectrogram from text, and the WaveRNN vocoder network transforms the mel spectrogram into time-domain waveform samples. These waveform samples are converted to audio, which is the output of our project. The adaptable modular design enables external users to quickly integrate the Text to Speech with Custom Voice system into their products. Additionally, users can edit specific modules and pipeline phases in this project without changing the source code. To achieve the best performance, the speaker encoder, synthesizer, and vocoder must be trained on a variety of speaker datasets.

Keywords: TTS, WaveRNN, Vocoder, Spectrogram

Sunitha N V
Assistant Professor
Department of Computer Science & Engineering
Mangalore Institute of Technology and Engineering
Moodabidri, Karnataka, India.
sunithanv6720@gmail.com

Suchetha N V
Assistant Professor
Department of Computer Science & Engineering
Sri Dharmasthala Manjunatheshwara Institute of Technology
Ujire, Karnataka, India.
itsmesuchethanv@gmail.com

I. INTRODUCTION

Speech synthesis technology has undergone a revolution thanks to the creation of the Text to Speech with Custom Voice system. This system has found use in a variety of fields, including accessibility, education, and entertainment thanks to its capacity to produce spoken recordings in unique voices. For example, the technology can be applied to produce voiceovers for videos, provide audio output for those who are blind or visually handicapped, and enhance the accessibility of digital information for users with various language or dialect requirements.

A WaveRNN vocoder, a synthesizer, and a speaker encoder make up the suggested system. Together, these modules create a unique voice that closely resembles the target speaker. The speaker encoder, which collects pertinent features from the speaker's speech and generates a fixed-dimensional embedding vector to represent the speaker's voice, is essential to the process. The synthesizer network based on Tacotron2 then uses this speaker embedding to extract a mel spectrogram from the input text. The mel spectrogram is then translated into time-domain waveform samples by the WaveRNN vocoder, which produces audio as the system's output.

The speech encoder, synthesizer, and vocoder need to be trained on various speaker datasets in order to operate at their peak levels. By doing this, the system is guaranteed to accurately reproduce the desired voice, even for speakers who were not part of the training set. Furthermore, the system's modular structure makes it simple for outside users to include it into their goods. The system is considerably more adaptable and adjustable because to the ability of users to update particular modules and pipeline phases without altering the source code.

1. **Problem Statement**: Research into text-to-speech technology with personalized voice cloning is now necessary due to the growing need for more personalized and natural audio content, as well as the development of a system that can clone voices without needing a lot of training data or retraining the model. Currently, training a deep neural network for voice cloning needs hours and hours of audio recordings of a single speaker, which can be expensive and time-consuming.

   By developing a system that can imitate voices from small samples of reference speech, we can lessen the amount of training data required and increase the efficiency of the voice cloning procedure. The user experience can be enhanced and more people can access digital information with the help of this technology. The capacity to execute voice cloning using only a little quantity of data can be useful in some situations when getting vast amounts of training data is not practical. For instance, this could be helpful if the speaker has died away or is no longer available for recording. Therefore, by exploring new deep learning methodologies and developing novel algorithms, we may improve the efficiency and accuracy of voice cloning systems, making it easier to create customized voices for a number of applications. These investigations are essential to the development of text-to-speech technology with personalized voice cloning.

2. **Objectives:** The main goal of this research is to develop a Text to Speech with Custom Voice system capable of producing speech audio in custom voices that are not included in

the training data. We want to create a system that can learn to produce speech patterns unique to individual speakers by utilizing the most recent developments in deep learning, giving users total control over the tone and manner of their voice.

Three main components—a speaker encoder, a synthesiser, and a WaveRNN vocoder—will make up the proposed system. The task of the speaker encoder is to examine the incoming speech data and produce a speaker embedding that captures the distinctive features of the target voice. The WaveRNN vocoder will convert the created audio into a waveform that can be played out through speakers or headphones. The synthesiser will use the speaker embedding to produce high-quality speech audio.

We want to produce a modular design that is simple to incorporate into a variety of goods and applications so that the suggested system is usable by outside users. Users will be able to benefit from the system's capabilities thanks to this modular architecture without having to spend a lot of time or money modifying their current software or hardware.

In addition to creating a cutting-edge Text to Speech with Custom Voice system, our goal is to research new deep learning techniques and create cutting-edge algorithms that can enhance voice cloning systems' efficacy and accuracy. These initiatives will concentrate on lowering the volume of recorded speech needed to train a model, making it simpler and more reasonably priced for users to produce customised voices for a number of applications.

3. **Scope:** This work intends to investigate the potential of a Text to Speech with Custom Voice system, which has a wide range of applications. The main goal of the system is to produce speech sounds in custom voices, even ones that are not included in the training data, by utilizing deep learning methods like the WaveRNN vocoder, speaker encoder, and synthesiser. Raw audio is transformed by the speaker encoder into a speaker embedding, which is then supplied into the voice synthesizer to create a unique voice. The unprocessed audio signal is then synthesized using the WaveRNN vocoder.

   Another crucial component of the system is its adaptable modular architecture, which enables third parties to swiftly incorporate the technology into their products. This can be especially helpful in fields where personalised voices can have a big impact, like entertainment, education, and accessibility. Custom voices, for instance, can be utilised to develop engaging interactive learning tools for students, especially those who struggle with learning. Custom voices can be used to create distinctive characters or voiceovers in the entertainment industry, bringing new levels of creativity and innovation.

   In order to improve the efficiency and precision of voice cloning systems, this work also intends to develop new deep learning techniques. A single speaker's recorded speech must be used for several hours in order to train a deep neural network for voice cloning, which can be both time-consuming and expensive. In order to make it easier to create customized voices for a variety of applications, we want to design new techniques that can duplicate sounds using fewer samples of reference speech.

II. LITERATURE REVIEW

1. **Literature Survey:** Several hours of speech from a single speaker are used to train a deep neural network in the present voice cloning method [1]. Although this method has proved effective in many text-to-speech applications, it has a number of limitations; including being time-consuming, expensive, and impractical in some situations, particularly when there is a lack of training data. Recent studies have suggested numerous unique strategies to enhance voice cloning systems and lessen the dependency on massive amounts of training data.

   Tacotron, an end-to-end voice synthesis model that can produce speech from characters, is one such method [1]. The model may be trained from scratch using pre-existing text and audio pairs and is based on a sequence-to-sequence architecture with attention. The authors separated the synthesis of the final waveform from the generation of the acoustic features by using a neural vocoder to convert the generated auditory characteristics into a waveform.

   An effective neural audio synthesis method dubbed WaveRNN was suggested in another work [2]. The authors proposed a new generation method based on subscale and used a weight-pruning methodology to minimise the WaveRNN's weights. Additionally recommended as a key component to enhance multi-speaker text-to-speech synthesis is transfer learning [3]. To extract speaker-specific features from audio signals, the authors of one study trained a speaker verification model on a substantial corpus of speaker verification data [4]. In order to enhance the effectiveness of voice cloning systems, they looked at the encoding layer and loss function in an end-to-end speaker and language recognition system.

   An end-to-end neural network architecture termed Char2Wav for speech synthesis from character-level text input was developed in another recent study [5]. In order to encode text input, the authors employed a convolutional neural network (CNN), and in order to produce speech, they used a WaveNet-like neural network. The model produced state-of-the-art outcomes in terms of voice quality and naturalness after being trained on a sizable dataset of single-speaker speech.

   Additionally, transfer learning from multi-speaker text-to-speech synthesis to speaker verification has been researched [3]. A approach for multi-speaker TTS synthesis employing a speaker verification model trained on a sizable dataset of clean speech was proposed by the authors of the paper [6]. They extracted speaker embeddings from the model and utilized those to condition a TTS system based on neural networks. Without the need for distinct training data for every speaker, the resulting system was capable of synthesizing speech in the voices of several speakers

Existing System: A system that duplicates voices accurately while using only a little quantity of reference speech can be created to get around the drawbacks of the existing.technique. This can be achieved by creating fresh deep learning methods and innovative algorithms. The user experience can be improved by using the suggested approach to personalize voices for a variety of applications. The suggested technique can also be useful in situations where gathering a substantial amount of training data is not

practical. The *suggested voice cloning technology will therefore be more effective, affordable, and usable than the current system.*

III. PROPOSED SYSTEM

1. **Limitations of Existing System:** The previous system's necessity for a substantial amount of training data, which may not always be available, is one of its major flaws. Furthermore, the current method might not work when the speaker is no longer accessible for recording or when the desired voice is not available. The inability of the existing voice cloning technique to create customised voices for a range of applications is another drawback.

2. **Proposed System:** A system that duplicates voices accurately while using only a little quantity of reference speech can be created to get around the drawbacks of the existing technique. This can be achieved by creating fresh deep learning methods and innovative algorithms. The user experience can be improved by using the suggested approach to personalize voices for a variety of applications. The suggested technique can also be useful in situations where gathering a substantial amount of training data is not practical. The suggested voice cloning technology will therefore be more effective, affordable, and usable than the current system.

IV. SYSTEM DESIGN

1. **Architecture to Proposed System:**

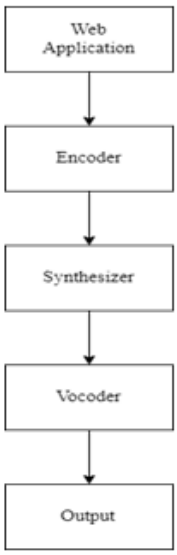

Figure 1: Architecture of the Proposed System

The process of defining the collection of hardware and software components and their interfacesto establish the framework for the development of the project. The figure 1 shows architecture diagram of the proposed system.

2. **Sequence Diagram:** A Sequence diagram is a type of interaction diagram because it describes how and in what order agroup of objects work together.

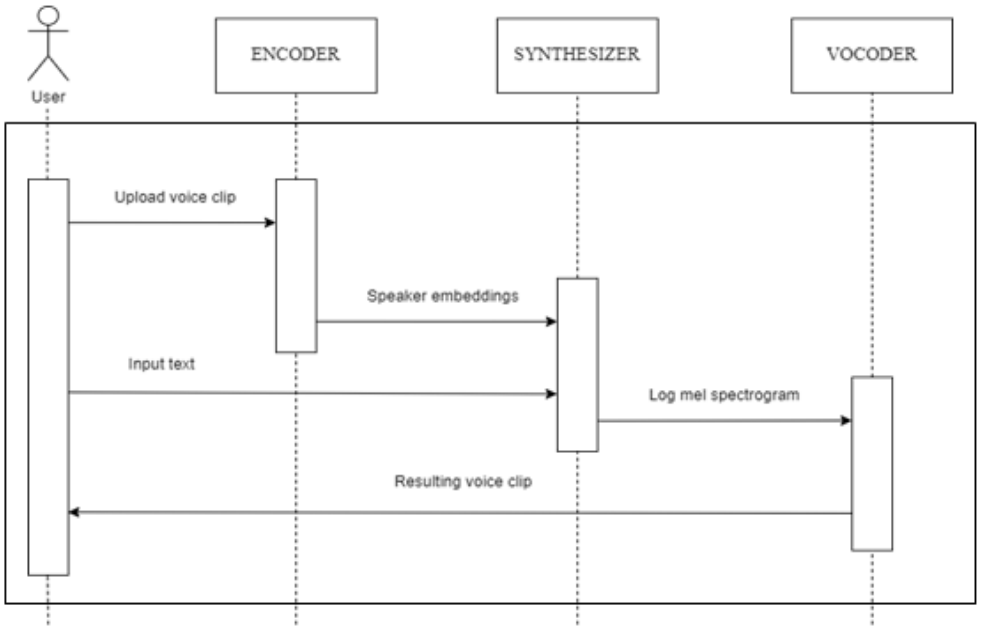

Figure 2: Sequence Diagram

3. **Dataflow Daigram:** A data flow diagram (DFD) is a graphical or visual representation using a standardized set ofsymbols and notations to describe a business's operations through data movement.

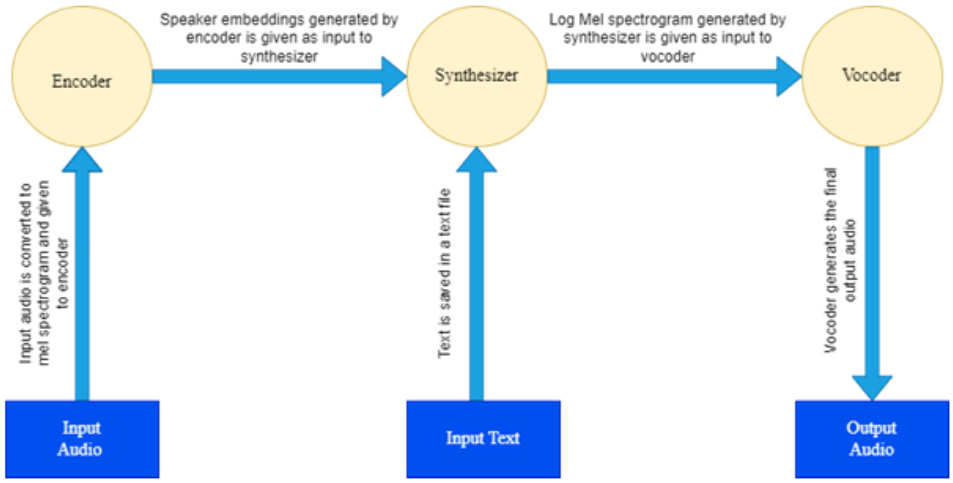

Figure 3: Dataflow Diagram

V. IMPLEMENTATION

1. **Module Implementation:** Three neural networks that were trained separately make up the system. The first network converts the speech signal into a fixed-dimensional vector and is in charge of encoding the speaker's voice. The second network is a synthesizer that uses

inputs like phonemes or graphemes to produce mel-spectrograms. Finally, the resulting spectrogram is transformed into time domain waves by the third network using an autoregressive WaveRNN vocoder. Together, these three networks create synthetic speech.

A brief reference utterance from the speaker to be copied is given to the speaker encoder at the moment of inference. A sentence that has been converted into a phoneme sequence is provided as input to the synthesizer, and it generates an embedding that is used to condition the synthesizer. The vocoder creates the speech waveform using the synthesizer's output. Figure 6.1 provides an illustration of this.

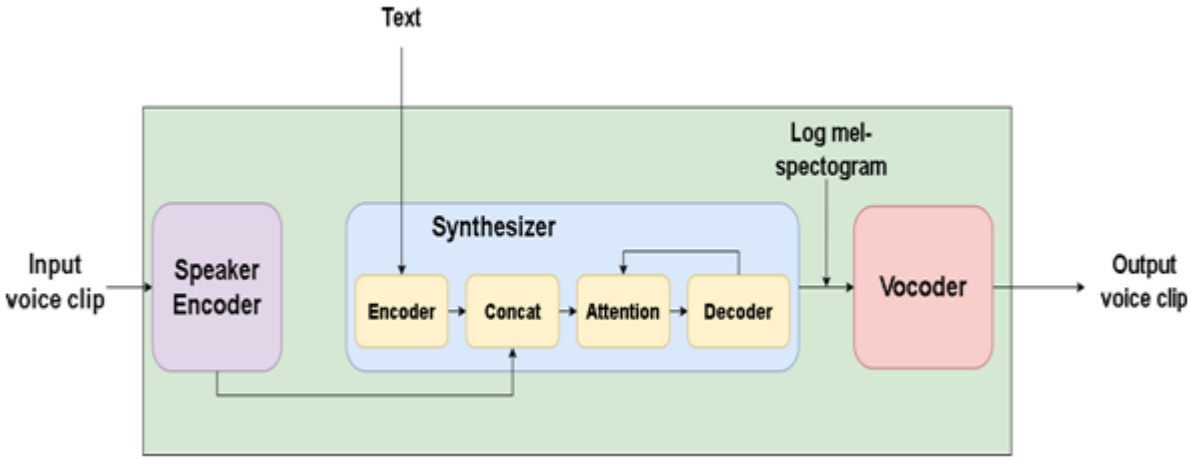

Figure 4: Components of Text-To-Speech with Custom Voice System

One feature is that each model may be trained independently on different datasets. One aims to have a noise-resistant model for the encoder that can capture the diverse qualities of the human voice. Therefore, it would be preferable to train the encoder on a wide corpus of numerous different speakers rather than imposing strict noise-level requirements. In addition, the encoder is trained using the GE2E loss, which only needs to know who the speaker is. When using GE2E, the model must learn a speaker verification task, which has little to do with voice cloning on its own. But the assignment is written so that the network will produce an embedding that accurately captures the speaker's speech.

The synthesizer can be trained on a voice using this embedding. Transcripts are necessary for the vocoder and synthesizer datasets, and the quality of the resulting audio can only be as excellent as the data. As a result, it is often the case that smaller, higher quality, and annotated datasets are needed.

2. **Speech Encoder:** A reference speech signal from the desired target speaker is used to condition the synthesis network using the speaker encoder. The usage of a representation that captures the features of various speakers and the capacity to recognize these qualities from just a brief adaption signal, independent of its phonetic content and background noise, are essential for effective generalization. A speaker-discriminative model that was trained on a text-independent speaker verification challenge satisfies these characteristics. The speaker encoder's goal is to create a latent, fixed-dimensional embedding that will be

utilized to train the synthesizer network using reference speech sounds from the selected target speaker. The latent embedding must capture all of the properties of the human voice, regardless of background noise or phonetic content, in order to achieve effective generalization across speakers.

For speaker verification, we use a neural network framework that is very scalable and precise. A series of log-mel spectrogram frames calculated from a voice utterance of any length are translated by the network into a fixed-dimensional embedding vector called the d-vector. The network is trained to maximize a generalized end-to-end speaker verification loss, resulting in high cosine similarity between embeddings of utterances from the same speaker and low cosine similarity between those of utterances from other speakers, which are spread out widely in the embedding space. There are no transcripts used; instead, the training dataset consists of 1.6-second-long segments of spoken audio examples with speaker identification labels.

Input A stack of three LSTM layers, each with 768 cells, is used to create the network, which receives 40-channel log-mel spectrograms. Each layer is then projected to 256 dimensions. The output of the top layer is L2-normalized at the last frame to produce the final embedding. An arbitrary length speech is divided during inference into 800 ms segments that are 50% overlapped. The outputs from each window's individual network runs are averaged and normalized to get the final utterance embedding. We discover that training on a speaker discrimination task results in an embedding that is directly suited for conditioning the synthesis network on speaker identity, even though the network is not directly optimized to learn a representation that captures speech attributes relevant to synthesis.

On a speaker verification task, the speaker encoder is trained. A common biometrics application is speaker verification, which uses a person's voice to confirm their identification. By determining a person's speaker embedding from a few utterances, a template is made for them. Enrollment is the name of this procedure. The system compares the embedding of a user's self-identification utterance with the enrolled speaker embeddings during runtime.

Each utterance in a training batch has a set duration of 1.6 seconds. These shorter entire utterances in the dataset were sampled from the greater complete utterances. Despite the fact that the model architecture can accommodate inputs of different lengths, it is reasonable to assume that it will work best with utterances that are similar in length to those used in training. As a result, a speech is divided into 1.6-second segments at inference time that overlap by 50%, and the encoder forwards each segment separately. To create the utterance embedding, the outputs are normalized and then averaged.

During inference, an arbitrary length speech utterance is divided into 1.6 second windows with a 50% overlap. The network then operates individually on each window, and the final utterance embedding is calculated by taking the average of all outputs. This method is illustrated in Figure 5

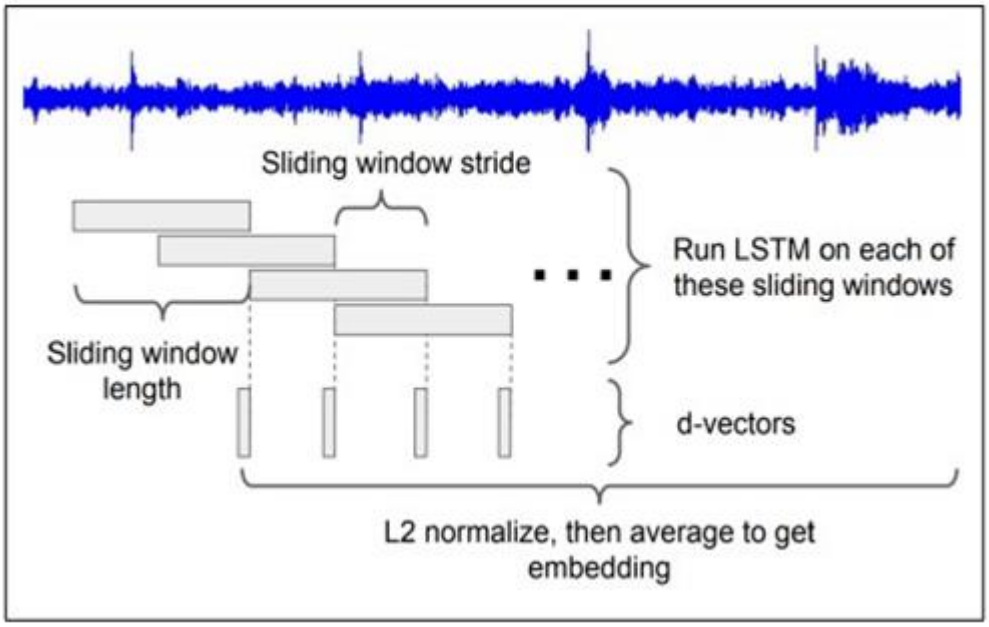

Figure 5: Sliding window approach used during inference

3. **Synthesizer:** To generate the Mel spectrogram from text, we employ a sequence-to-sequence neural network based on Tacotron 2, which is conditioned on the speaker embedding. At each time step,the speaker embedding is concatenated with the synthesizer encoder output to condition the outputand match the speaker's voice.

The architecture features an attention mechanism that connects a recurrent sequence-to-sequence encoder-decoder architecture. Modifications to the architecture are made to allow multiple speakers in a manner akin to Deep Voice 2. To transfer the characteristics of the target voice, the speaker embedding vector acquired from the Speaker encoder network, which corresponds to the target speaker, must be merged into the Synthesizer network. At each time step, the embedding vector and encoder output of the synthesizer network are combined. For the model to converge across different speakers, it is sufficient to feed the concatenated embeddings to the attention layer.

Mel-frequency spectrogram, a low-level acoustic representation, is used as an intermediate feature representation. This format is smoother than waveform representations and is easily generated from time-domain waveforms. It is simpler to train with a squared error loss since it is phase invariant within each frame. Mel-spectrograms provide a more compact summary of the frequency data. It is calculated using a non-linear transform applied to the short-time Fourier transform (STFT) frequency axis, which was motivated by how the human auditory system works. Humans do not hear audio frequencies on a linear scale, according to study. Lower frequency variations are easier to spot than higher frequency variations. The Mel-scale is an example of an auditory frequency scale that emphasizes lower-frequency aspects that are essential for speech intelligibility while downplaying subtleties in higher frequencies that are mostly dominated by fricatives and other noise bursts and do not require high-fidelity modeling.

An encoder-decoder network connected by attention makes up the architecture of the synthesizer. A latent feature representation is created by the encoder using the input character sequence. We concatenate the speaker embedding to each encoder frame rather than just using the encoder output for decoding. This makes it easier for the synthesis process to add the desired voice qualities to the resulting mel-spectrograms.

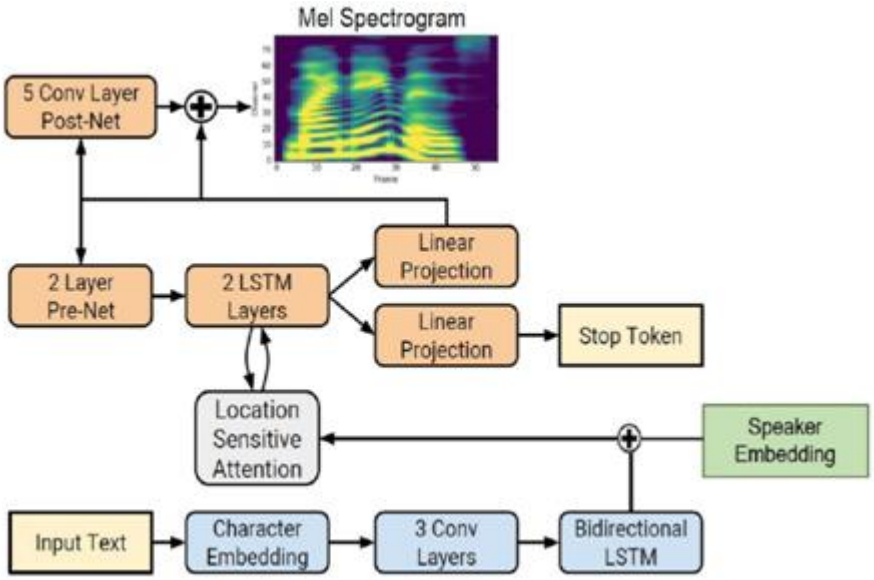

Figure 6: Tacotron2 architecture conditioned on speaker-embeddings

The blue and orange blocks represent the encoder and the decoder, respectively. The decoder then employs this concatenated form to forecast a spectrogram. The input characters are represented by a learnt 512-dimensional character embedding, which is subsequently processed by a stack of three convolutional layers, batch normalization, and activation. Each convolutional layer includes 512 filters with a 5 x 1 shape and a 5 character range. The convolutional layers model the long-term context (like N-grams) associated with the input character sequence. The output of the final convolutional layer is fed into a single 512-unit bi-directional LSTM layer, or 256 units in each direction, to create the encoded features.

The synthesizer's decoder uses an autoregressive recurrent neural network to predict a mel-spectrogram one frame at a time from the encoded input sequence. For the prediction produced by the previous time-step, an information bottleneck layer consisting of a pre-net with two fully connected layers and 256 hidden units with ReLU is utilised. The pre-net output and the attention context vector are combined, then fed via a stack of two unidirectional LSTM layers, each with 1024 units. The concatenation of the LSTM output and the attention context vector is projected through a linear transform to forecast the target spectrogram frame. Finally, the predicted mel spectrogram is fed into a convolutional post-net with 5 layers, which determines a residual to add to the prediction and improve the entire reconstruction. Each post-net layer is made up of 512 filters with a shape of 5 x 1. All layers except the final layer aresubjected to batch normalization and tanh activations.

4. **Vocoder:** The system uses WaveRNN as a vocoder, which performs autoregressive synthesis to create time-domain audio waveforms from Mel spectrograms produced by the synthesis network. By training on input from numerous speakers, the synthesizer network is able to easily create a multi-speaker vocoder by capturing all the features required for high-quality synthesis of distinct voices in the form of Mel spectrograms.

The 60 WaveNet convolutions are all replaced by a single GRU layer in WaveRNN. The ground truth audio serves as the model's target and the GTA mel spectrogram produced by the synthesizer serves as its input. The model makes predictions for fixed-size waveform segments during training. The bottom 8 bits (coarse) of the target 16-bit sample are predicted first and are then utilized to condition the prediction of the higher 8 bits in the forward pass of WaveRNN, which is implemented with only N = 5 matrix-vector products. The output is sampled from a distribution from which parameters are being predicted.

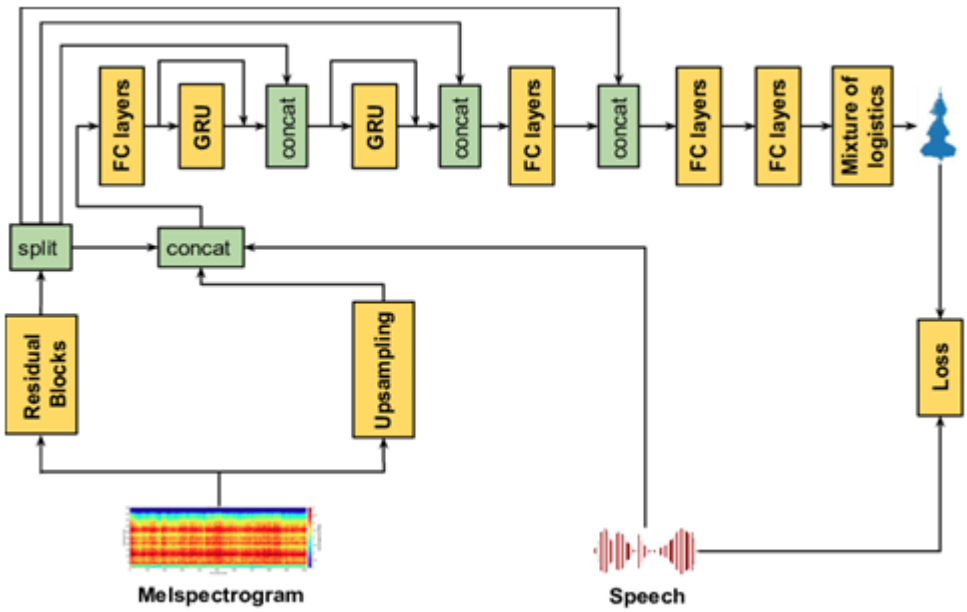

Figure 7: The WaveRNN architecture.

As shown in Figure 7 this design consists of residual block sand up sampling network, followed by GRU and FC layers. The conditional network and the recurrent network are the two main networks that make up the architecture. A pair of residual networks, an up-sampling network, and a network with three scaling factors make up the conditioning network. With the use of several residual blocks, we first transfer the input's acoustic properties, or mel-spectrograms, to a latent representation. After that, the latent representation is divided into four components, each of which will serve as an input to the recurrent network. To match the desired temporal size of the input signal, the up-sampling network is put into place. Speech is supplied into the recurrent network along with the outputs of these two convolutional networks, namely the residual and up-sampling networks. Two unidirectional GRUs are used in the recurrent network, and there are a few fully connected (FC) layers at the very end. By design, the overhead complexity is decreased by using fewer parameters and better utilizes the temporal context.

The speaker encoder's output is not directly conditioned on by the network. A multi-speaker vocoder can be created by simply training on data from numerous speakers since the mel spectrogram predicted by the synthesizer network captures all of the pertinent information required for high quality synthesis of a variety of voices.

VI. TESTING

1. **Testing Levels:** Testing is part of Verification and Validation. Testing plays a very critical role for qualityassurance and for ensuring the reliability of the software.

The objective of testing can be stated in the following ways.

- A successful test is one that uncovers as-yet-undiscovered bugs.
- A better test case has high probability of finding un-noticed bugs.
- A pessimistic approach of running the software with the intent of finding errors.

Testing can be performed in various levels like unit test, integration test and system test.

- **Unit Testing:** To guarantee that the different parts function properly, unit testing is used. Each component is examined separately from the rest of the system. Each module of this system was properly tested, and the results were compared to the predicted output. Unit testing concentrates verification work on the software design module's tiniest unit.
- **Integration Testing:** Another type of testing is integration testing, which is typically carried out to find issues with how data moves between interfaces. Since the unit-tested components are grouped and tested in short segments, it is simpler to identify and fix mistakes. This strategy is maintained until all modules have been integrated to create the entire system.
- **System Testing:** System testing is a type of testing that verifies a fully integrated, finished piece of software. A system test's objective is to assess the complete system requirements. The software is typically just a small part of a bigger computer-based system. The software is ultimately interfaced with other software and hardware systems. System testing is basically a collection of several tests performed solely for the goal of exercising the entire computer-based system.
- **Acceptance Testing:** Acceptance Testing is a method of software testing where a system is tested for acceptability. The major aim of this test is to evaluate the compliance of the system with the business requirements and assess whether it is acceptable for delivery or not. After testing the system at different levels, how the complete system is accepted which meets all the mentioned functional and non-functional requirements.

2. **Test Cases:** A test case is a software testing document, which consists of events, action, input, output, expected result and actual result. Technically a test case includes test description, procedure, expected result and remarks. Test cases should be based primarily on the software requirements and developed to verify correct functionality and to establish conditions that reveal potential errors. Individual PASS/FAIL criteria are written for each test case. All the tests need to get a PASS result for proper working of an application.

Table 1: Test cases for Project

Test Numbers	Test Case	Expected Results	Actual Output	Status
1	Running the Web Application.	The Web application should run on local host on webbrowser	As expected, result.	Pass
2	Taking audio input from the user in wav or flac format.	The Web application should take audio input from the device.	As expected, result.	Pass
3	Taking text input from the user in .txtformat.	The Web application should take text file input from the device.	As expected, result.	Pass
4	Generating the output audio file.	The web application should generate anaudio filein the target speaker's voice for the given text.	As expected, result.	Pass

VII. RESULTS AND ASNAPSHOTS

Initially, when the user visits the website's homepage, they will be presented with the choice to input audio in either .wav or .flac format, as well as the option to upload a text file in .txt format.

Figure 8: Home page of the website

Once the user has provided the audio and text file inputs, they can proceed by clicking the submit button. After clicking the submit button, the user will receive a message indicating thatthe audio file is being generated. The generation of the audio file will take some time.

Figure 9: Result generation page

Figure 9: Output Audio Page

Upon uploading the audio and text file and clicking the submit button, the application will commence generating the audio file. Once the audio file has been generated, it will be playedback to the user on the output page.

The Mean Opinion Score (MOS) metric is a widely accepted method to measure the quality of synthesized speech. In this study, we used MOS to evaluate the synthesized speech of ten speakers, five of whom were included in the training data and five who were not. The rating scale used ranged from 1 (Least Similar/Natural) to 5 (Extremely Similar/Natural) in increments of 0.5.

Our evaluation yielded a Naturalness rating of 4.1, which indicates that the synthesized speech is highly natural and similar to human speech. A Naturalness score of 4.1 suggests that there are very few noticeable artifacts or distortions in the speech, making it sound like it was produced by a human speaker. This is a particularly promising result as the speakers used in the evaluation include both trained and untrained voices, indicating that our model can generate high-quality speech for a wide range of speakers.

In addition to the Naturalness rating, our evaluation also yielded a Similarity rating of 3.7, which indicates that the synthesized speech is moderately similar to the original speech. While the Similarity rating is not as high as the Naturalness rating, it still suggests that our model is capableof producing speech that is relatively close to the original input.

Overall, our results suggest that the model we developed is highly effective at synthesizing natural-sounding speech for a wide range of speakers. This is a promising result, as natural-sounding synthesized speech has a range of applications, including text-to-speech systems, voice assistants,and audio books.

The evaluation was done on a rather limited sample size of 10 speakers, it should be noted. Although the results are encouraging, it is crucial to evaluate the model using a bigger sample size in order to more thoroughly verify its efficacy. Further investigation into the precise elements that affect the Naturalness and Similarity ratings may also be done in order to provide guidance for future model development.

VIII. CONCLUSION AND FUTURE WORK

In conclusion, the development of a text-to-speech system using personalised voice cloning technology has opened up new possibilities for creating distinctive and realistic-sounding voices. The availability of large speech datasets and advancements in machine learning techniques have made it possible to create high-quality customised voices that closely match the voice of a human speaker. Applications for Text To Speech with Custom Voice System include greater accessibility to digital content, more personalized audio content delivery, and improved human-computer interactions. However, there are still a number of challenges in creating a real-time voice cloning system that can produce speech in real-time without any audible pauses or errors.

Future research could concentrate on a number of topics to improve the functionality of the voice cloning system. One of the key areas that require improvement is the system's ability to produce speech that sounds more realistic. Customers may currently find it difficult to interact with the information because the system's speech can sound artificial or mechanical. Researchers may look into incorporating more NLP techniques into the system to address this issue and produce expressive, fluid speech.

The system could be improved further by adding support for more languages. The systemnow only supports English, which reduces the number of users it could accommodate. The technology could benefit a larger audience by enhancing its language skills to incorporateadditional spoken languages, such as Indian languages.

Additionally, future studies can concentrate on creating a system that is lighter and more effective and can be included into a variety of applications. The system's computational complexitymight be reduced, and it could be made to work better in real-time, making it an important tool forimproving the accessibility and customization of digital content.

REFERENCES

[1] W. Ping, K. Kainkaryam, Y. Wang, D. Stanton, Y. Zhang, R. J. Weiss, and B. Li. (2018). TACTRON: Towards end-to-end speech synthesis. arXiv preprint arXiv:1703.10135.
[2] K. Arık, A. H. van den Oord, and O. Vinyals. (2019). Efficient Neural Audio Synthesis. In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, California, USA, 9-15 June 2019 (pp. 232-241).
[3] Y. Zhang, Y. Fan, N. Hu, and L. Zhang. (2019). Transfer Learning from Speaker Verification to Multi-Speaker Text-To-Speech Synthesis. arXiv preprint arXiv:1806.04558.
[4] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur. (2018). Exploring the Encoding Layer and Loss Function in End-to-End Speaker and Language Recognition System. In Proceedings of the 19th Annual Conference of the International Speech Communication Association (INTERSPEECH 2018) (pp. 306-310).
[5] H. Zen, A. Senior, and M. Schuster. (2019). Char2Wav: End-to-end speech synthesis. arXiv preprint arXiv:1905.02316.
[6] Jia, Y., Zhang, Y., Weiss, R. J., Wang, Q., Shen, J., Ren, F., Chen, Z., Nguyen, P., Pang, R., Moreno, I. L., & Wu, Y. (2019). Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6965-6969). IEEE.

