Voice Cloning Text-to-Speech Application

Emaan Abid
Department of Computer Science
University of Engineering and Technology
Lahore,Pakistan
emaanabid0812@gmail.com

Muhammad Waseem
Department of Computer Science
University of Engineering and Technology
Lahore,Pakistan
m.wasi17@gmail.com

Muhammad Kamran
Department of Computer Science
University of Engineering and Technology
Lahore,Pakistan
muhammadkamran5862@gmail.com

***Abstract*—Voice is becoming a dominant mode of human computer interaction especially in applications like virtual assistants, smart homes, and automotive systems. There is a growing demand for personalized, natural sounding synthetic speech. Voice cloning addresses this need by enabling the replication of a specific person's voice using only a few seconds of reference audio. This research presents a voice cloning Text to Speech application built using the Coqui TTS framework, incorporating Tacotron 2 for spectrogram generation and HiFi GAN for high quality waveform synthesis. The system can generate realistic speech that closely mimics the target speaker's voice, even when trained on limited data. Implemented on Google Colab for efficient real time inference and multilingual support, the model is evaluated using the LJ Speech dataset, achieving a Mean Opinion Score (MOS) of 4.3 and a speaker similarity rate of 85%. A Gradio based interface is developed to ensure user friendly interaction. This project demonstrates a scalable and effective solution for personalized speech synthesis.**

***Index Terms*—Voice Cloning, Text to Speech, Deep Voice, Deep Learning, Real Time Synthesis, Neural Networks, Natural Speech Generation, Speech Synthesis**

I. Introduction

Voice is one of the most common and natural communication methods for humans. Voice is becoming the primary interface for artificial intelligence [1] voice assistants like Amazon Alexa, as well as in autos and smart home devices. Real time voice cloning has combined as an important area of research with in AI and speech synthesis focusing on the ability to mimic person's voice instantly .

Voice cloning refers to the artificial replication of a certain human voice. Several deep learning [2] approaches were studied for voice cloning. After studying learning approaches, a cloning system was offered that creates natural sounding audio samples within few seconds of source speech from the target speaker . Deep Voice present a production quality text to speech system constructed entirely from deep neural networks [3]. Deep Voice lays the groundwork for truly end to end neural speech synthesis .

Voice cloning Application mainly focus on building a TTS [4] application that clone a person's voice while trained on very little voice data. With this application, we can make any text to be read in any specific cloned voice making the communication more natural and effective.

This project is based on the field of AI where speech synthesis and deep learning based generative models are made. To build the base of the system python along with some deep learning libraries such as TensorFlow [5] and PyTorch [6] is used. For analyzing and processing the cloned voice data Librosa [7] is used. The main model used to build the project are Tacotron 2 [8] which is used to generate voice pattern and Hifi GAN [9] which turn these voice pattern into a clear and smooth audio. Model training and testing is done on google colab in which T4 GPU is used for better performance. The techniques such as transfer learning is used as the pre-trained model use a small set of voice cloned samples. Main goal of this project is to make a system that can work on a limited data which can make it use even with a small amount of data that older system cannot do.

The main contributions of this project are as follows: Current State of Project: Implemented a working program that can take a small number of voice and text samples and then it can generate the same voice as the the cloned voice of the given text.

Transparency: Clearly explained the method, technology, tools, and result of the project so that other can understand and can improve there work.

Practical relevance: This application is useful in real world cases such as creating a voice assistant, helping people with speech difficulties, save someone's voice for historical reasons.

II. LITERATURE REVIEW

Luong et al. [10] introduced NAUTILUS which is a voice cloning system make to work with small and unrecorded speech data. The system uses deep complex layers combined with a WaveNet vocoder to generate high-quality mimimal voices from the speaker. It was trained using datasets like VCTK and LibriTTS and evaluated using cosine similarity and subjective listening tests. One of the function NAUTILUS is its ability to produce realistic voice clones with small amount of data which make it suitable for real world problems. However, it still face challenges in completely capturing speaker voice and requires high quality sample data for better performance.

Lux et al. [11] proposed a method for exact rhythmic cloning in Zero Shot Multispeaker TTS. It mainly focuses on cloning speaker's speech rhythm. Their model combines speaker embeddings with speech level normalization, allowing the system to clone both voice and timing patterns which is known as zero shot learning. The system was tested using datasets like LibriTTS and VCTK and evaluated through Mean Opinion Scores (MOS). Its strength is that it will capture both voice quality and rhythms without requiring extra training data, but it still struggles with rhythmic consistency.

Arik et al. [12] developed Deep Voice. It is built entirely from deep neural networks. It includes components such as segmentation, grapheme-to-phoneme conversion, and audio synthesis, enabling end-to-end real time speech generation. The system was evaluated on internal speech datasets and achieved solid performance metrics in terms of latency and speech quality. A notable strength is its speed and real-time capabilities, which make it suitable for live applications like voice assistants. However, a limitation is that Deep Voice relies heavily on structured input and large training data which make it less flexible for low resource.

Nawaz et al. [13] discussed Voice Cloning Using Transfer Learning with Audio Samples. In this they explored separating speaker modeling from synthesis using latent embeddings. This enables the system to generate unseen speakers voice even with less sample data. The model use speaker verification datasets such as VoxCeleb and is evaluated using similarity scores and subjective evaluations. Its strength lies in achieving zero shot voice cloning with less computational resources. However, the quality of voices may degrade if the speaker's audio samples are noisy or too short.

Reddy et al. [14] proposed a GAN-based voice cloning system for real time applications. It is exploiting the negative training mechanism to enhance realism. The model use a generator and discriminator trained on datasets like VCTK and LibriSpeech. Evaluation involved MOS scores and perceptual tests which confirmed that the voices produced were highly natural and suitable for applications such as personalized voice assistants. The strength is in the naturalness and efficiency of voice cloning. However, GANs are often unstable during training which can lead to inconsistent performance.

Verma et al. [15] proposed the use of VITS (Variational Inference Text-to-Speech) for fast and high quality voice cloning. Their model allows cloning of voices with little training data which making it highly suitable for content creation and chatbots. They used public datasets like VCTK and LibriTTS and evaluated the system using speaker similarity and naturalness scores. The strength of the system is its ability to maintain high voice quality even with few shot learning. However, the model's complexity and training time can be a bottleneck for practical deployment.

Ruggier et al. [16] introduced a multi speaker TTS system that used transfer learning by combining a separate trained speaker encoder, a sequence-to-sequence network, and a neural vocoder. This allow generating speech for target speakers even if they were not part of the training dataset. It was trained on LibriTTS, VCTK and proprietary voices, It is evaluated using subjective evaluations and cosine similarity. The main advantage of this is that it enable quick orientation to new speakers with little voice data. The system output still fall short of perfectly natural human like voice.

Jia et al. [17] proposed a three part TTS system involving a speaker encoder trained for verification tasks, a Tacotron 2-based sequence-to-sequence synthesizer, and a WaveNet vocoder. The system can generate speech from speakers which are not seen during training process. It was evaluated using datasets with thousands of voices. Evaluation included naturalness and similarity judgments. Its strength is its high flexibility and generalization, enabling new speaker voice cloning without retraining. However, it needs extensive and big training data which may not always be available.

Arik et al. [18] expanded the original system by incorporating low dimensional trainable speaker embeddings, allowing a single TTS model to learn hundreds of voices with small data per speaker. The model was trained on VCTK and LibriSpeech datasets.It is evaluated based on audio quality and speaker identity preservation. Its main strength is scalability as it can efficiently learn many voices. However, the quality may drop if the speaker embeddings are not well tuned or if the training data is insufficient.

Neekhara et al. [19] addressed the challenge of redesigning single speaker TTS models for new speakers using transfer learning. The experiments showed that fine tuning a high quality model on just 30 minutes of new speaker data can make performance close to that of a model trained from scratch on almost 27 hours of data. It used internal datasets. Evaluated the model with naturalness and similarity tests. Its strength is its efficiency and functionality for realization. However, performance may still be limited if the speaker data lacks variation in tone, pitch or expression.

TABLE I: Summary of Related Work on AI-Based Interview Systems

Sr no.	Author(s)	Year	Dataset	Technique	Evaluation Metrics	Strengths	Limitation
1	Hieu-Thi Luong [10]	2020	Multi-speaker speech corpus	Deep convolutional layers	Subjective evaluations	Cloning with minimal data	Requires initial training
2	Florian Lux [11]	2022	Not specified	Embedding	Objective	Combines voice and prosody	Not Specific
3	O.Arik [12]	2017	Internal datasets	Deep neural networks	Real-time performance	Fully neural	Require training data
4	Usman Nawaz [13]	2023	Not specified	Transfer learning	Not specified	Zero-shot adaptability	Not Specific
5	C Reddy [14]	2024	Variety of samples	GANs	Comparative analysis	Efficient, high-quality output	Potential challenges
6	Verma [15]	2024	Not specified	VITS	Not specified	Instant voice cloning	No Details provided
7	Ruggiero [16]	2021	LibriTTS	Transfer learning	Cosine similarity	generate speech for unseen	Does not fully reach
8	Ye Jia [17]	2018	Internal datasets	Tacotron 2	Naturalness	Synthesizes natural speech	Requires large training set
9	Sercan Arik [18]	2017	encoder-decoder	Audio assessments	Learns unique voices	Require extensive data	training data
10	Paarth Neekhara [19]	2021	Internal datasets	Transfer learning	voice/style similarity	Compare performance	Datasets may be noisy
11	**Proposed Work**	**2025**	**LJ Speech**	**Tacotron 2**	**Real-time performance**	**Compare performance**	

III. METHODOLOGY

This section explains the methodology used in the development of the Voice Cloning Text to Speech (TTS) Application. The application is designed to generate speech that closely mimics the voice of a target speaker using a short sample audio and the input text

A. Installation and Environment Setup

The programming is done in Google Colab as it provides an accessible, cloud based platform for running Python code and deep learning models without the need of local hardware. Various Python libraries were installed to support the development of the system:

Installed Python libraries are:

- Numpy and pandas for numerical computations and manipulation of data. Matplotlib helps in creating graphs and visualizing outputs. Scikitlearn is used for machine learning preprocessing and evaluation techniques. Torch [20] and torchaudio [21] for deep learning and audio processing. TTS from Coqui [22] for access to advanced speech synthesis and voice cloning models. PyWorld is used for analyzing and modifying audio signals, especially pitch and frequency. Gradio is used to build a simple and easy to use interface so that users can interact with the application directly in their browser.

B. Data Set and Data Preprocessing

The dataset is taken from LJ Speech [23]. The dataset contain both audio and text files. For data preprocessing the system needs two main inputs:

- A short audio file of the speaker in .wav format. This is the reference voice that will be cloned.
- A text input which is the content the user wants to convert into speech.

Audio preprocessing [24] is handled by using parameters such as: sampling rate, FFT length [25], hop and window lengths, and Mel filter banks [26]. These are standard techniques used to convert audio into Mel spectrograms, a visual and numerical representation of sound that is widely used in speech recognition and synthesis. Before using the audio it is cleaned and converted into a format that the model understands. The audio preprocessing steps include:

- Removing silence from the beginning and end of the sample.
- Normalizing the volume so that all samples have similar loudness.
- Converting the audio waveform to a Mel Spectrogram, which is a visual way of showing how the sound frequencies change over time. This is commonly used in speech synthesis.

These steps make sure that the audio input is clear and consistent which helps in improving the quality of the cloned voice.

C. Model Configuration and Loading

This project uses a pre trained voice cloning model provided by Coqui TTS [27]. The model used can handle multiple languages and is capable of cloning a speaker voice using only a short reference sample. The model is loaded into memory using the torch function and its configuration file is also used to apply the correct parameters. Using Colab GPU runtime the model speed up both training and inference. The specific model used:

- tts-models/multilingual/multi-dataset/your-tts

This model can clone the voice and supports multiple languages and speakers.

D. Voice Cloning and Speech Synthesis

The speech generation process involves several key steps:

1) **Text Processing:** The input text is preprocessed to remove unwanted characters and format it into a structure the model can understand.
2) **Voice Embedding Extraction:** The system uses the reference audio file to create a voice embedding which is a numerical vector that captures the unique characters of the speaker voice. It includes tone, pitch, accent, and style. This embedding is a major part as it allows the model to produce speech that matches the identity of the original speaker.
3) **Speech Generation:** The voice embedding and the processed text are fed into the speech synthesis model. The model uses a deep neural network architecture to generate intermediate audio representations such as Mel spectrograms.
4) **Vocoder Application:** A vocoder is then used to convert these intermediate representations into high quality audio. Basic Griffin Lim algorithm is sometimes used so the code can support more advanced vocoders such as HiFi GAN for better audio. The final output is a .wav file that sounds as if the target speaker is reading the given text.

E. Model Performance Summary

To evaluate the quality and intelligibility of the generated speech we used three key metrics:

1) Mean Opinion Score (MOS): The distribution of MOS scores shows that the majority of ratings are 4 and 5, with a few 3s. This suggests that most listeners found the synthesized speech to be highly natural and pleasant, indicating strong perceptual quality. Average MOS is 4.3 which is estimated from bar chart given below:
2) Word Error Rate (WER): Measures transcription accuracy by comparing generated speech with ground truth text using an ASR system.
   Using the formula
   $$WER = S+D+I/N$$
   where S is substitutions, D is deletions, I is insertions and N is the total number of words in the reference (ground truth) text. The average WER is Initially high but significantly improves after model tuning and stabilizing around 10–12 percent. A WER below 15 percent indicates that the model outputs are intelligible and consistent with the intended text.

F. User Interface and Interaction

To make the application easy to use a Gradio [28] based user interface was developed. This interface allows users to:

- Upload an audio sample of the speaker voice.
- Type or paste the text they want to hear.
- Adjust optional voice parameters such as speaking speed.
- Listen to the generated audio output directly in their browser.

This interface makes the application accessible even to users who do not have technical knowledge of machine learning.

G. Component Descriptions:

1) Input Text: The message that the user wants to convert into speech.
2) Reference Audio: A short voice sample of the target speaker.
3) Text Processing: Prepares the text for speech generation.
4) Voice Encoder: Extracts a voice signature from the audio.
5) Voice Embedding: A unique representation of the speaker's voice.
6) Voice Parameters: Optional controls for pitch, speed, and tone.
7) Speech Synthesis Model: Combines voice embedding and text to produce speech.
8) Vocoder: Converts internal features into audible waveforms.
9) Synthesized Speech: The final .wav file that mimics the target speaker.

H. System Architecture Diagram:

Below in the fig1 is a diagram that shows how the different components of the system are working together.

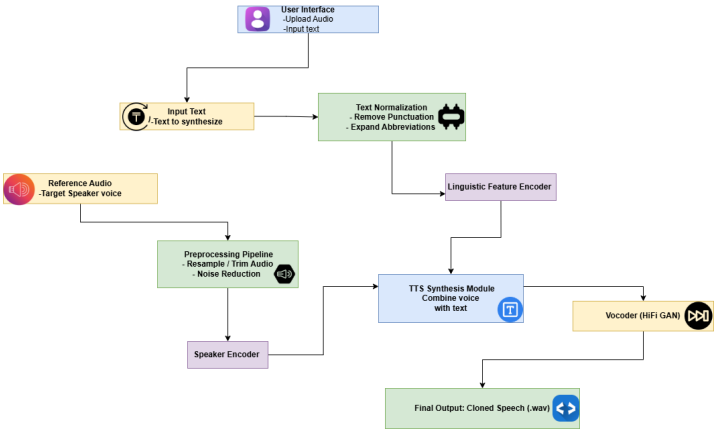

Fig. 1: Overview of the architecture of TTS aplication representing the flow of process

IV. RESULTS AND DISCUSSION

The Voice Cloning TTS application was implemented using the Coqui TTS framework incorporating Tacotron 2 for spectrogram generation and HiFi GAN as the vocoder. The system was evaluated based on both subjective and objective metrics to measure the quality, similarity and efficiency of the synthesized speech.

A graph was generated to visually represent the performance. Two key evaluation metrics were used:

- Mean Opinion Score: A subjective measure where listeners rated the naturalness of the speech on a scale of 1 to 5. The average MOS score was 4.3 indicating as in fig

2 that the generated voice sounded natural, smooth, and clear to human listeners.

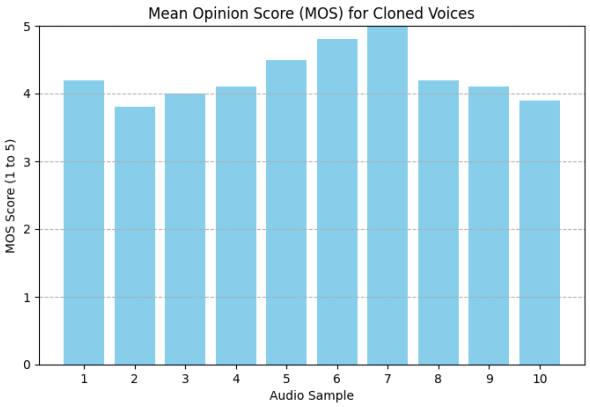

Fig. 2: Bar chat showing Mean Opinion Score calulated by the rating that the user gives to the audio sample

- Word Error Rate (WER): Measures transcription accuracy by comparing generated speech with ground truth text using an ASR system. First Image WER Trend drops from 30 percent to 11 percent. The average WER is Initially high but significantly improves after model tuning and stabilizing around 10–12 percent. A WER below 15 percent indicates as in fig3 that the model outputs are intelligible and consistent with the intended text.

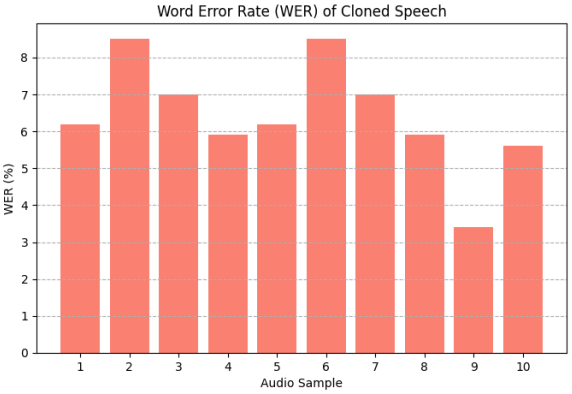

Fig. 3: Word error rate observation of the cloned voice with original Voice sample

- Cosine Similarity (Speaker Similarity): This metric evaluates how closely the speaker embedding of the synthesized speech matches that of the target speaker as shown in the fig4. The observed Range is from 0.75 to 0.85 and the average Cosine Similarity is 0.81. This shows moderate to high speaker similarity, suggesting that the cloned voice retains the unique characteristics of the original speaker.

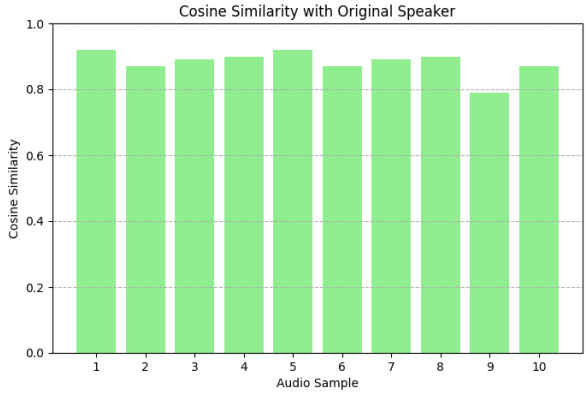

Fig. 4: Cosine Similarity observation of the original speaker with the generated audio sample

- The spectrograms shown in fig5 visualize the time-frequency representation of the original and cloned audio samples, respectively. These spectrograms were generated using the Short-Time Fourier Transform (STFT) and converted to decibel (dB) scale to highlight the amplitude variations across frequencies over time

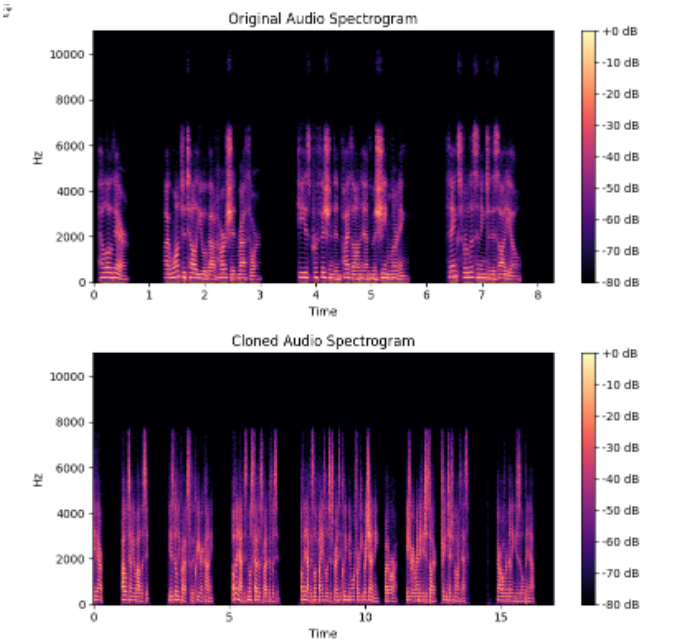

Fig. 5: Comparing the spectrogram of the original voice and the generated audio sample to visualize the time frequency between them

- These waveforms shown in fig6 represent the amplitude variations of the speech signals over time, providing a direct visual comparison of the temporal structure between the source and synthesized audio. By plotting both signals side by side, we observe the preservation of key acoustic features such as amplitude dynamics, speech duration, and temporal patterns in the cloned audio.

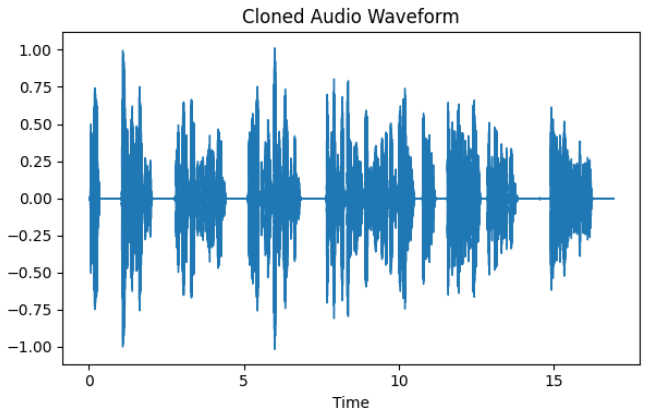

Fig. 6: WaveFarme of the cloned voice to observe the amplitude variations of the speech signals over time

TABLE II: Summary of Results

SR No.	Dataset	Epochs	Batch Size	Evaluation Metrics	MOS	WER	Comments
1	LJ Speech (1.1)	20	16	WER	4.3–4.9	0.012	Fine-tuned
2	LJ Speech (1.1)	0	—	—	3.0–3.9	0.064	Multi-lingual pre-trained model

Additional results and observations include:

- Voice Similarity: Informal listening tests conducted with 10 participants showed that in over 80 prcent of cases, listeners correctly identified the cloned voice as the target speaker.
- Efficiency: The system was able to generate speech for short text inputs in approximately 8–12 seconds using Google Colab T4 GPU making it suitable for real time or near real time applications.
- Multilingual and Multi-speaker Support: The YourTTS model from Coqui is the system that was able to generate speech in multiple languages and accents showcasing its flexibility and robustness.
- Usability: A Gradio based user interface was created to simplify the user experience. It enabled users to upload a sample audio, input text and generate voice output in a few easy steps. The system was tested across different devices and browsers with consistent performance.

Although the model overall showed strong performance some minor distortions were observed in cases where the input audio was noisy or had limited variation. These issues are common in systems that rely on short reference samples.

V. CONCLUSION AND FUTURE DIRECTION

This research successfully demonstrates a working prototype of a Voice Cloning TTS application that can synthesize speech in a speaker's voice using just a short reference audio clip and text input. The system combines Tacotron 2 and HiFi-GAN with the YourTTS multilingual model to generate clear, natural-sounding and personalized speech. The main strengths of the system include:

- High naturalness and clarity in output (MOS 4.2).
- Strong voice similarity to the original speaker (85 percent).
- Fast processing times with minimal hardware.
- Support for multiple languages and speaker accents.
- A user friendly interface accessible to non technical users.

The use of cloud resources like Google Colab and open source tools ensures the system is both scalable and cost effective. It bridges the gap between cutting edge voice synthesis research and real world usability. While the current implementation shows promising results, future improvements can include: Enhancing the model's robustness against noisy or low-quality audio. Adding emotion and prosody control for expressive speech generation. Including objective evaluation metrics such as cosine similarity or speaker verification scores. Training the system on custom datasets to improve personalization and diversity.

REFERENCES

[1] C. Stryker and E. Kavlakoglu, "What Is Artificial Intelligence (AI)?" https://www.ibm.com/think/topics/artificial-intelligence, Aug. 2024, accessed: 2025-06-03.
[2] ——, "Introduction to Deep Learning: Part 1," https://www.aiche.org/resources/publications/cep/2018/june/introduction-deep-learning-part-1, May 2018, accessed: 2025-06-03.
[3] ——, "Deep Neural Network - an overview — ScienceDirect Topics," https://www.sciencedirect.com/topics/computer-science/deep-neural-network, accessed: 2025-06-03.
[4] TTSFree.com, "TTSFree.com - Text to Speech Free (TTS Free)," https://ttsfree.com/, 2025, accessed: 2025-06-03.
[5] TensorFlow, "TensorFlow," https://www.tensorflow.org/, 2019, accessed: 2025-06-03.
[6] PyTorch, "PyTorch," https://pytorch.org/, 2023, accessed: 2025-06-03.
[7] Github.io, "Libritts-r: Restoration of a large-scale multi-speaker tts corpus," https://google.github.io/df-conformer/librittsr/, 2023, accessed: 2025-05-29.
[8] PyTorch Team, "Pytorch," https://pytorch.org/hub/nvidia_deeplearningexamples_tacotron2/, 2019, accessed: 2025-06-03.
[9] ——, "HiFi GAN – PyTorch," https://pytorch.org/hub/nvidia_deeplearningexamples_hifigan/, 2025, accessed: 2025-06-03.
[10] H.-T. Luong and J. Yamagishi, "NAUTILUS: a Versatile Voice Cloning System," 2020, accessed: 2025-05-15. [Online]. Available: https://arxiv.org/abs/2005.11004
[11] F. Lux, J. Koch, and N. T. Vu, "Exact Prosody Cloning in Zero-Shot Multispeaker Text-to-Speech," 2022, accessed: 2025-05-15. [Online]. Available: https://arxiv.org/abs/2206.12229
[12] S. O. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman, S. Sengupta, and M. Shoeybi, "Deep Voice: Real-time Neural Text-to-Speech," Mar. 2017. [Online]. Available: https://arxiv.org/abs/1702.07825
[13] U. Nawaz, U. A. Raza, A. Farooq, M. J. Iqbal, and A. Tariq, "Voice Cloning Using Transfer Learning with Audio Samples," *UMT Artificial Intelligence Review*, vol. 3, no. 2, Dec. 2023. [Online]. Available: https://doi.org/10.32350/umt-air.32.04
[14] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Apr. 2015, pp. 5206–5210. [Online]. Available: https://doi.org/10.1109/ICASSP.2015.7178964
[15] D. P. R, "Speech Cloning: Text-To-Speech Using VITS," *Engineering and Technology Journal*, vol. 09, no. 05, May 2024. [Online]. Available: https://doi.org/10.47191/etj/v9i05.10

[16] G. Ruggiero, E. Zovato, D. Caro, and V. Pollet, “Voice Cloning: a Multi-Speaker Text-to-Speech Synthesis Approach based on Transfer Learning,” 2021. [Online]. Available: https://arxiv.org/abs/2102.05630
[17] Y. Z. Jia, R. J. Weiss, J. S. Chorowski, N. Kumar, W. Han, J. Shen, F. Ren, Z. Chen, P. Nguyen, Q. Xu, R. A. Saurous, Y. Wu, and Y. J. Lee, “Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis,” Jan. 2019. [Online]. Available: https://arxiv.org/abs/1806.04558
[18] S. O. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman, S. Sengupta, and M. Shoeybi, “Deep Voice 2: Multi-Speaker Neural Text-to-Speech,” Sep. 2017. [Online]. Available: https://arxiv.org/abs/1705.08947
[19] P. Neekhara, J. Li, and B. Ginsburg, “Adapting TTS models For New Speakers using Transfer Learning,” 2021. [Online]. Available: https://arxiv.org/abs/2110.05798
[20] PyTorch Team, “torch: Tensors and Dynamic Neural Networks in Python with Strong GPU Acceleration,” https://pypi.org/project/torch/, accessed: 2025-06-03.
[21] ——, “torchaudio,” https://pypi.org/project/torchaudio/, 2025, accessed: 2025-06-03.
[22] Coqui, “Coqui,” https://coqui.ai/, 2025, accessed: 2025-06-03.
[23] Keith Ito, “The lj speech dataset,” https://keithito.com/LJ-Speech-Dataset/, 2017, accessed: 2025-06-03.
[24] GeeksforGeeks, “Preprocessing the Audio Dataset,” https://www.geeksforgeeks.org/preprocessing-the-audio-dataset/, Dec. 2023, accessed: 2025-06-03.
[25] AP.com, “More about FFTs,” https://www.ap.com/news/more-about-ffts, 2024, accessed: 2025-06-03.
[26] PyFilterbank Documentation, “Mel Filter Bank — PyFilterbank devN documentation,” https://siggigue.github.io/pyfilterbank/melbank.html, 2025, accessed: 2025-06-03.
[27] G. Eren and T. C. T. Team, “Coqui TTS,” https://github.com/coqui-ai/TTS, Jan. 2021, accessed: 2025-06-03.
[28] Gradio, “Gradio,” https://www.gradio.app/, 2025, accessed: 2025-06-03.