ADVANCING VOICE CLONING FOR NEPALI LANGUAGE: LEVERAGING TRANSFER LEARNING IN LOW-RESOURCE LANGUAGE

Manjil Karki [1] *, **Pratik Shakya**[2], **Ravi Pandit**[3], **Sandesh Acharya**[1], **Dinesh Gothe**[1]

[1] Department of Computer Engineering, Khwopa College of Engineering, Tribhuvan University

Abstract

Voice cloning refers to synthesizing speech that mimics the vocal characteristics of a specific individual using a limited number of audio samples. This technology finds extensive application in areas such as personalized voice interfaces, assistive technologies, and digital content creation. However, most of the existing voice cloning systems are developed on high-resource languages that have an upper hand on extensive annotated datasets. In contrast, this study introduces a novel voice cloning framework specifically designed for the Nepali language, a low-resource language with limited linguistic and acoustic resources. The proposed system uses a combination of a speaker encoder, a Tacotron2-based synthesizer, and a WaveNet vocoder, trained through a transfer learning approach leveraging multilingual pre-trained models to mitigate the challenges caused by data scarcity. To support this effort, we constructed a dataset of a Nepali speech corpus comprising 168 hours of audio data from 546 speakers and adapted the entire synthesis pipeline to accommodate the Devanagari script and the phonological nuances of the Nepali language. Evaluation through both subjective and objective metrics demonstrates the system's effectiveness, with mean opinion scores (MOS) of 3.93 for naturalness and 3.29 for speaker similarity, as well as a low equal error rate (EER) of 0.005. These results affirm the feasibility of achieving high-quality voice cloning in low-resourced language contexts and establish a robust foundation for further exploration and development in Nepali speech synthesis and voice cloning.

Keywords: Voice cloning, Low-resource language, Nepali speech synthesis, Transfer learning, Speaker encoder, Tacotron2, WaveNet

1. INTRODUCTION

The creation of synthetic speech imitators using advanced artificial intelligence (AI) algorithms that are indistinguishable from real voices is known as voice cloning. It is commonly associated with terms like artificial voice, speech creation, and deepfake audio. Voice cloning is a customised process as opposed to TTS (text-to-speech) systems, which use a pre-existing technology to translate text into spoken words. It recognises and makes use of specific vocal characteristics for different speech patterns. In the past, TTS used two techniques: parametric TTS, which used statistical models but produced less realistic-sounding outcomes, and concatenative TTS, which relied on recorded audio but lacked emotion. Currently, AI and Deep Learning enhance synthetic speech quality, leading to the widespread use of TTS applications, ranging from phone systems to virtual assistants such as Siri and Alexa (Daspute et al., 2020; Zhang and Lin, 2022).

Voice cloning is the technology employed to generate artificial voices that mimic particular people, utilized in entertainment and support (Neekhara et al., 2021). TTS systems have transitioned from primitive rule-based approaches that generated mechanical speech to contemporary systems employing machine learning methods, like deep neural networks, which allow for more lifelike audio. Significant progress encompasses technologies such as Google's Tacotron (Jia et al., 2018).

Obstacles persist, especially in creating multilingual TTS systems because of the scarcity of non-English speech data and ethical issues related to privacy and the risk of abuse, including the production of counterfeit audio record-

*Corresponding author: Manjil Karki
Khwopa College of Engineering, Tribhuvan University
Email: manjilkarki2000@gmail.com
(Received: March 28, 2025 Accepted: June 4, 2025)
https://doi.org/10.3126/jsce.v12i1.82362

ings. Notwithstanding these challenges, advancements have been achieved in TTS technology, with continuous work required to improve linguistic intricacy and tackle ethical concerns.

The evolution of TTS includes early rule-based systems, concatenative synthesis, statistical parametric speech synthesis, and the current use of deep neural networks (Gibiansky et al., 2017), all contributing to the improvement of high-quality speech synthesis.

2. Methodology

2.1. Proposed Method:

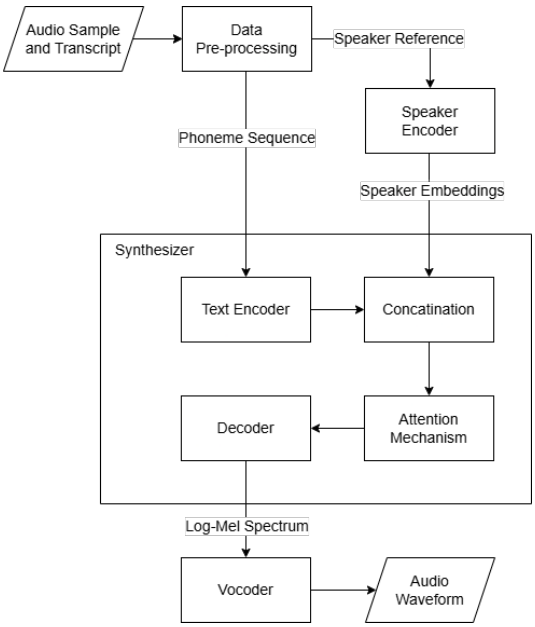


Figure 1. Block diagram of Nepali voice cloning

The proposed method in Figure 1 of voice cloning consists of three major models with initial preprocessing of data files (Jemine et al., 2019).

2.2. Dataset Creation

The dataset was obtained via the open-source platform OpenSLR (Kjartansson et al., 2018; Sodimana et al., 2018). The collection contains a total of 168.34 hours of audio and corresponding transcripts in Devanagari script. It was then transformed further into audio multiple files of length 5-20 seconds practising proper data augmentation. At the end of dataset preparation we had about 154,235 pairs of speech and its transcript files. In terms of speakers count on gender, the dataset was more female-biased, which we tried to balance out by data augmentation accordingly.

```
<corpus root>
    |
    .- train-clean-100/
                  |
                  .- 19/
                      |
                      .- 198/
                      |    |
                      |    .- 19-198.trans.txt
                      |    |
                      |    .- 19-198-0001.flac
                      |    |
                      |    .- 14-208-0002.flac
                      |    |
                      |    ...
                      |
                      .- 227/
                           | ...
```

Figure 2. Speech transcript pairs for training

The sample dataset for speech transcript pairs for training is shown in Figure 2.

2.3. Data Pre-Processing

Data preprocessing plays a crucial role, as each of the three core models—encoder, synthesizer, and vocoder—require a distinct pipeline. For the encoder, preprocessing involves extracting raw audio samples from the dataset and converting them into encoded mel-spectrogram representations. The synthesizer pipeline, on the other hand, integrates audio files, transcripts, and corresponding utterances to generate a comprehensive dataset. This includes mel-spectrograms, audio-spectrograms, speaker embeddings, and metadata files containing relevant textual information. Subsequently, the processed mel-spectrograms are transformed into a Ground-Truth Aligned (GTA) dataset, which serves as input for vocoder preprocessing. As illustrated in Figure 3, the complete pipeline includes normalization of textual input and conversion of audio data into mel-scale spectrograms and speaker-specific embeddings.

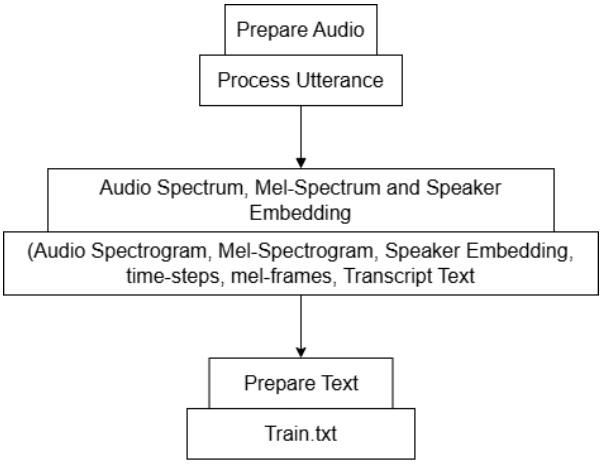


Figure 3. Synthesizer model data pre-processing

2.4. Encoder:

The encoder plays a crucial role in deriving compact and informative representations—commonly referred to as embeddings—from input speech data. Its primary function is to encapsulate the unique traits and vocal features of a speaker into a fixed-dimensional vector (Arik et al., 2018). While various architectural configurations can be employed, a widely adopted approach involves the use of a Mel-spectrogram-based encoder (Chen et al., 2018). This type of encoder generally comprises multiple convolutional layers integrated with batch normalization and nonlinear activation mechanisms, such as the Rectified Linear Unit (ReLU). The process begins by transforming the raw audio waveform into a Mel-spectrogram, which illustrates the temporal evolution of the signal's spectral characteristics. The extracted Mel-spectrogram is then passed through convolutional layers to identify key features, which are later condensed into a fixed-length embedding vector—typically using pooling methods or recurrent neural networks (RNNs).

2.5. Synthesizer:

Tacotron2 (Shen et al., 2018) is a widely used architecture for text-to-speech synthesis. The architecture comprises a text encoder that transforms the input text into a fixed-dimensional embedding, followed by a decoder that produces mel-spectrograms as the acoustic representation of speech. The encoder combines convolutional layers with bidirectional RNNs to capture linguistic features. The decoder uses an autoregressive setup with stacked LSTM or GRU layers (Fan et al., 2014) and attention mechanisms to produce spectrograms. During training, the model minimizes a loss function to align generated outputs with target spectrograms. Tacotron2 is known for its ability to generate high-quality synthetic speech.

2.5.1 Architecture

The Tacotron2 text-to-speech (TTS) system uses a deep neural network composed of the following key components (Wang et al., 2017):

- **Text Encoder:** Converts input text (characters or phonemes) into hidden representations using convolutional layers and a bidirectional RNN. These embeddings capture the semantic meaning of the input.
- **Attention Mechanism:** Aligns text and audio by learning attention weights that determine which parts of the input text are most relevant at each decoding step.
- **Decoder:** Generates spectrogram frames one at a time using the attention context and previously generated frames. It uses an RNN and incorporates attention outputs to create acoustic features.
- **Post-Processing Network:** Converts the decoder's spectrogram output into a waveform using techniques like Griffin-Lim and applies further signal processing to enhance audio quality.

2.5.2 Alignment Plots

Alignment plots are visual tools used in speech and language processing to visualize the correspondence between two sequences, such as audio and its transcription (Helander et al., 2008). Typically displayed as heat-maps, scatter plots, or line plots, these plots map one sequence (e.g., audio) on the x-axis and another (e.g., predicted transcription) on the y-axis. They help identify errors, evaluate system performance, and guide improvements in tasks like speech recognition and machine translation.

2.5.3 Mel-Spectrogram

A Mel spectrogram (Habib et al., 2021) is a time-frequency representation of an audio signal where frequencies are scaled according to the Mel scale, reflecting human auditory perception. It is generated by segmenting the signal, applying a Fourier transform, and mapping the resulting spectra through a Mel filter bank. Commonly used in speech recognition and audio analysis, Mel spectrograms emphasize perceptually important frequencies, making them effective input features for machine learning models while also reducing data dimensionality.

2.6. Vocoder:

WaveNet, (Van Den Oord et al., 2016), is a prominent vocoder architecture extensively used in voice cloning applications. It models raw audio through conditional probability distributions using dilated convolutional neural networks (CNNs). When provided with a mel-spectrogram as input, WaveNet synthesizes the corresponding audio waveform in a sequential, sample-by-sample manner. The architecture's use of stacked dilated convolutions allows it to capture long-range temporal relationships, making it especially effective in generating speech that is both natural and perceptually realistic.

2.7. Transfer Learning:

Transfer learning aims to enhance model generalization and address data scarcity by leveraging knowledge from a source domain (Mei et al., 2021). Initially, deep learning models were trained from scratch, but performance was limited due to poor data quality and high training complexity. To overcome these challenges, transfer learning was applied to all three models. While a suitable pre-trained model for

the Nepali language and voice was unavailable, a multilingual model was used instead. This approach yielded significantly better results than training from scratch, though it required additional fine-tuning to achieve satisfactory performance.

3. Result

3.1. Training Analysis

During training, the primary results observed were charts and plots illustrating the model's progression. While loss and accuracy curves are typically key indicators, this audio-based model places greater emphasis on subjective evaluation, as standard metrics alone do not fully capture all relevant aspects of performance.

3.2. Encoder Training

For encoder training, some of the key parameters that were observed are listed as follows:

3.2.1 Encoder Training Loss

The encoder's training loss is defined by an end-to-end objective function that evaluates the quality of speaker clustering in the UMAP (Uniform Manifold Approximation and Projection) space. This is optimized using the GE2E (Generalized End-to-End) loss, which encourages embeddings from the same speaker to cluster tightly while pushing embeddings from different speakers apart.

Training batches consist of multiple speakers, each with several utterances. Specifically:

- Let N be the number of distinct speakers in the batch.
- Let M be the number of utterances per speaker.
- Let e_{ij} denote the embedding of the j-th utterance from the i-th speaker, where $1 \leq i \leq N$ and $1 \leq j \leq M$.

To represent each speaker, the model computes a *speaker centroid* or *speaker embedding*, denoted as c_i, by averaging the embeddings of that speaker's utterances.

$$c_i = \frac{1}{M} \sum_{j=1}^{M} e_{ij} \tag{1}$$

This centroid in Equation (1) represents the overall voiceprint of speaker i. The GE2E loss then uses these centroids and individual utterance embeddings to calculate similarity scores and optimize the model to produce compact, well-separated speaker clusters.

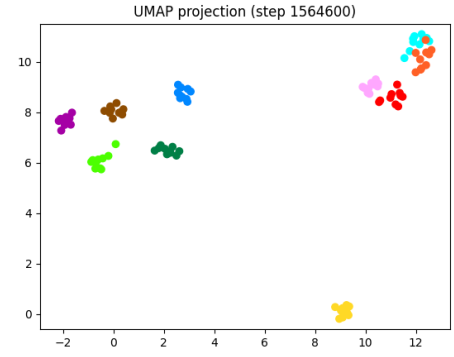
3.2.2 Equal Error Rate for Encoder

In biometric authentication systems, one of the most important indicators of performance is the Equal Error Rate (EER). This metric identifies the threshold at which the system's likelihood of incorrectly accepting an unauthorized user matches the likelihood of incorrectly rejecting a legitimate one. Specifically, it balances the rates of false acceptances and false rejections, offering a single value that reflects the trade-off between the two. Lower EER values signify improved precision in verifying user identities. Table 1 illustrates the encoder training procedure and the associated loss.

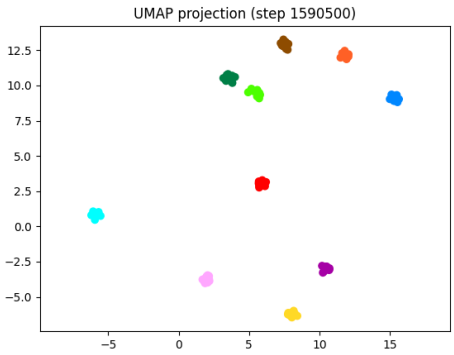
Table 1. Encoder training results

Parameters	Results
Encoder Training Loss	0.02 ± 0.01
Equal Error Rate	0.005 ± 0.001

3.2.3 UMAP Projection



(a) Initial Cluster



(b) Cluster After Training

Figure 4. U-Map Projection at Different Stages of Training

UMAP (Uniform Manifold Approximation and Projection) (Jiale and Ying, 2020) is a dimensionality reduction method

designed for visualizing and clustering high-dimensional data. It excels in capturing nonlinear relationships, making it suitable for complex data distributions. UMAP projects the data into a two-dimensional space, enabling effective visualization as shown in Figure 4. It has been successfully applied across various domains, including image processing, genomics, and natural language processing.

3.3. Synthesizer Training

3.3.1 Attention Scaled Dot Product

In Transformer models, attention is a mechanism that allows the model to weigh the importance of different input tokens when processing a sequence. A commonly used method for computing attention is the Scaled Dot-Product Attention, which efficiently captures the relationships between tokens using vector similarity.

The attention computation is given by Equation (2):

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{2}$$

Symbols

- Q: Query matrix — represents the set of query vectors. Each query corresponds to a token trying to attend to other tokens in the sequence.
- K: Key matrix — contains key vectors used to compute similarity scores with queries.
- V: Value matrix — contains the actual values to be aggregated using attention weights.
- d_k: Dimensionality of the key vectors — used to scale the dot product for numerical stability.

Working

1. Compute the dot product $QK^\top$ to measure similarity between queries and keys.
2. Scale the result by $\frac{1}{\sqrt{d_k}}$ to prevent large values that could dominate the softmax.
3. Apply the `softmax` function to obtain attention weights (a probability distribution).
4. Multiply the attention weights with the value matrix V to obtain the final output.

This attention mechanism enables the model to dynamically focus on relevant parts of the input sequence during processing.

3.3.2 M1 Loss

Mean Absolute Error (MAE), also referred to as L1 loss or M1 loss, Measures the average absolute difference between predicted and actual values; less sensitive to outliers.

The mathematical expression for MAE is given in Equation (3).

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3}$$

Symbols

- n: Denotes the total count of data samples in the dataset.
- y_i: Represents the ground truth or actual value corresponding to the i-th observation.
- $\hat{y}_i$: Indicates the model's predicted value for the i-th sample.
- $|y_i - \hat{y}_i|$: Captures the magnitude of the deviation between the predicted and actual values for the i-th instance.

3.3.3 M2 Loss

Mean Squared Error (MSE), also known as L2 loss or M2 loss, Calculates the average of squared differences between predicted and actual values; emphasizes larger errors more.

The mathematical expression for MAE is shown in Equation (4).

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4}$$

Symbols

- n: The total count of samples within the dataset.
- y_i: The true value corresponding to the i-th data point.
- $\hat{y}_i$: The estimated value predicted for the i-th data point.
- $(y_i - \hat{y}_i)^2$: The squared error between the true and predicted values for the i-th data point.

3.3.4 Final Loss

In regression tasks, it is common to combine M1 loss (Mean Absolute Error) and M2 loss (Mean Squared Error) to leverage the strengths of each—balancing robustness to outliers with sensitivity to large errors. MAE calculates the average absolute difference between predictions and targets, providing robustness against outliers. In contrast, MSE computes the average squared difference, which places greater emphasis on larger errors. The combined loss function integrates these properties, balancing sensitivity to large errors with overall robustness, thereby improving both accuracy and generalization. The combined loss is defined as in Equation (5)

$$\text{Loss} = \text{MAE} + \text{MSE} \tag{5}$$

Components

- (Mean Absolute Error): Measures the average magnitude of errors between predicted and actual values, offering greater resistance to outliers.
- (Mean Squared Error): Captures the average squared error; emphasizes larger deviations more strongly.
- Loss: The total combined loss that incorporates both absolute and squared differences.

This combined loss can help improve a model's ability to generalize by simultaneously accounting for both the direction and magnitude of prediction errors. Table 2 illustrates loss and errors during training of synthesizer

Table 2. Synthesizer training

Parameters	Results
Mean Absolute Error	0.2930 ± 0.0010
Mean Squared Error	0.0495 ± 0.0001
Total loss (MAE + MSE)	0.3525 ± 0.0100

3.3.5 Mel-spectrogram

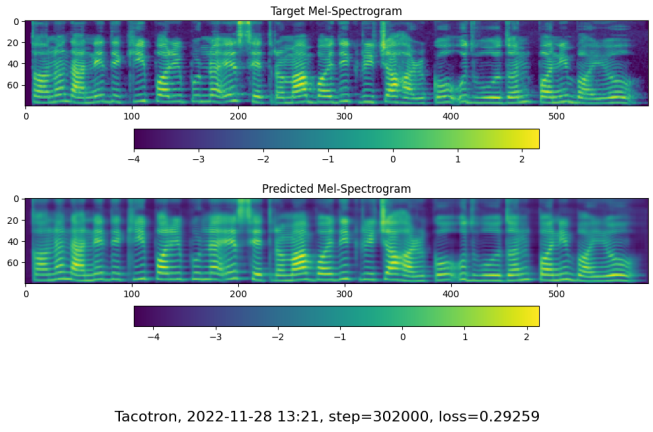


Figure 5. Mel-spectrogram comparison: Synthesizer training

Figure 5 illustrates target and predicted mel-spectrogram during training.

3.3.6 Alignment

Figure 6 illustrates learned alignment by the attention mechanism.

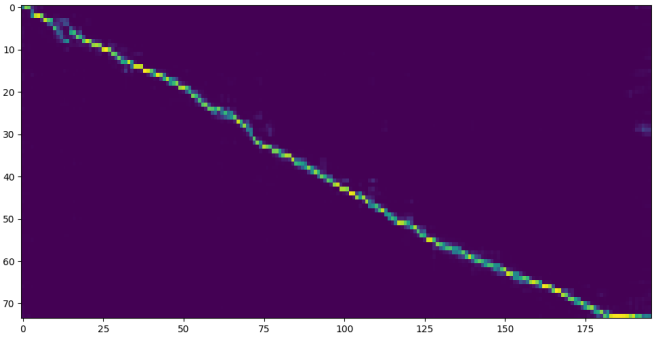


Figure 6. Alignment heat-map: Synthesizer training

3.4. Vocoder

Training a vocoder is a supervised process done with pairs of synthesized mel-spectrogram from synthesizer and raw target audio waveform. Table 3 represents the loss during the vocoder training process.

Table 3. Results of vocoder training

Parameter	Result
Vocoder Training Loss	4.095 ± 0.005

3.5. Metrices

A range of tools, techniques, and evaluation metrics were utilized during model training for analysis and selection. Similar evaluation techniques were employed to ensure the reliability and validity of the final models and outcomes.

3.5.1 Mean Opinion Score

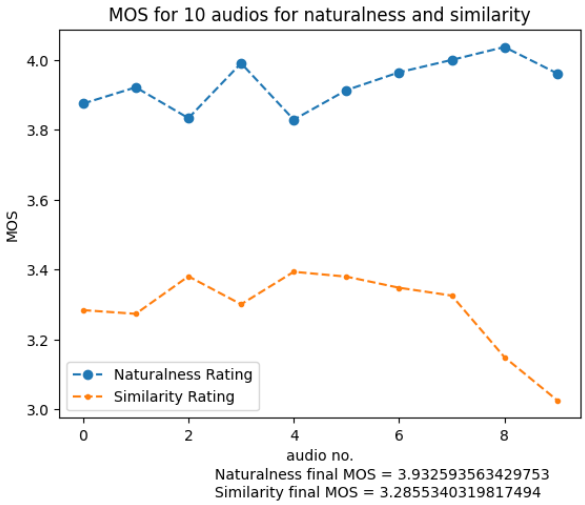


Figure 7. MOS (Mean Opinion Score)

The Mean Opinion Score (MOS) is a widely used subjective metric to assess the perceived quality of audio or video signals by human listeners or viewers ("Mean Opinion Score

(MOS) Revisited: Methods and Applications, Limitations and Alternatives", n.d.). In this study, MOS was derived from responses of 124 participants—50 familiar with the project and 70 unfamiliar—who rated 10 cloned and real voice samples via a Google Form on a scale of 1 to 5 for naturalness and similarity. The collected data was cleaned using Python and pandas, where rows with more than six NaN values were removed, and others were imputed with the mean. After cleaning, data from 120 participants were retained, and average scores for each sample were computed as shown in Figure 7.

Table 4 presents the Mean Opinion Score (MOS) for Naturalness and Similarity, evaluated on ten generated audio samples.

Table 4. Naturalness and similarity in terms of MOS

	Results
Audio Property	MOS
Naturalness	3.93
Similarity	3.29

3.5.2 PESQ

Perceptual Evaluation of Speech Quality (PESQ) is a standardized metric used to assess speech quality in telecommunications by comparing a degraded signal to a clean reference signal through a perceptual model that reflects human hearing (Rix et al., 2001). The algorithm generates scores ranging from -0.5 to 4.5, typically mapped to a MOS scale of 1 (bad) to 5 (excellent). In this project, PESQ as shown in Figure 8 was computed for cloned voice samples from both the training and test datasets, yielding scores of 2.8 on validation data and 2.3 on test data. These relatively low scores are attributed to the limited quality of the training dataset.

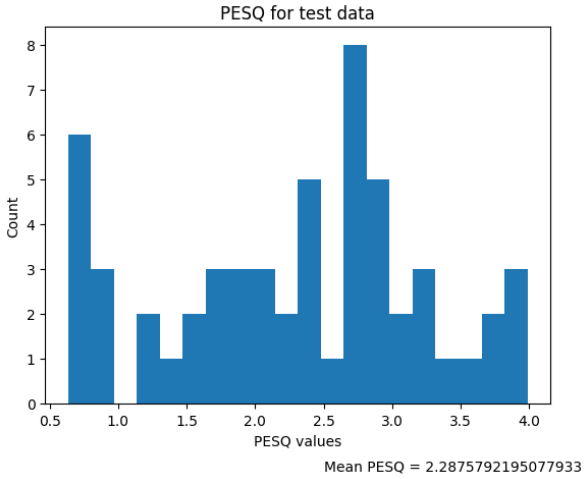


Figure 8. PESQ (Perceptual evaluation of speech quality)

3.5.3 U-MAP

Similar to the projection applied during training, embeddings for both the original and cloned voices were computed and visualized after dimensionality reduction. Figure 9 presents the Uniform Manifold Approximation and Projection plot, with multiple clusters, each cluster is represents a unique user. Each cluster is a collection of dots and crosses; dots represent original audio, and crosses represent generated audio. The proximity of original and generated samples within clusters indicates the model's effectiveness and high accuracy in preserving speaker characteristics.

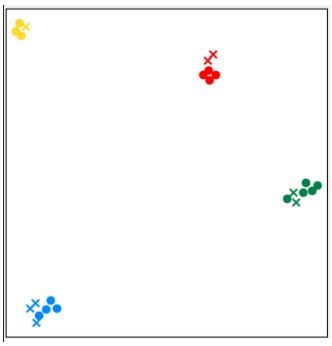


Figure 9. UMAP of multiple users and cloned audio

4. Conclusion

In conclusion, the proposed Nepali Voice Cloning system focuses on replicating voices with Nepali-specific accents and dialects, using input text in the Devanagari script. The system is structured around three main components: the encoder, synthesizer, and vocoder. The speaker encoder employs a convolutional neural network (CNN) followed by a recurrent neural network (RNN) to convert a user's voice into a unique speaker embedding—a form of vocal fingerprint used to distinguish between speakers. This embedding, together with the input text in Devanagari script, is passed to the synthesizer. The synthesizer, a modified version of the Tacotron architecture, consists of an encoder, attention mechanism, and decoder, and is adapted to incorporate speaker embeddings. It produces a mel-spectrogram, which is then fed into the vocoder to generate the corresponding audio waveform. All components are deployed within a web application, allowing the complete system to function seamlessly in an integrated environment.

Quantitatively, the system performs reasonably well, achieving a Mean Opinion Score (MOS) of 3.9 for naturalness and 3.2 for speaker similarity on a scale from 1 to 5. Additionally, the Perceptual Evaluation of Speech Quality (PESQ) score was 2.3 on the validation set and 1.8 on the test set, with PESQ scores ranging from -0.5 to 4.5.

References

Arik, S., Chen, J., Peng, K., Ping, W., & Zhou, Y. (2018). Neural voice cloning with a few samples. *Advances in Neural Information Processing Systems*, *31*.

Chen, Y., Assael, Y., Shillingford, B., Budden, D., Reed, S., Zen, H., Wang, Q., Cobo, L. C., Trask, A., Laurie, B., et al. (2018). Sample efficient adaptive text-to-speech. *arXiv preprint arXiv:1809.10460*.

Daspute, K., Pandit, H., Chandran, S. N., et al. (2020). Real time voice cloning: Voice converter using DeepSpeech and Tacotron.

Fan, Y., Qian, Y., Xie, F.-L., & Soong, F. K. (2014). TTS synthesis with bidirectional LSTM based recurrent Neural Networks. *Interspeech*, 1964–1968.

Gibiansky, A., Arik, S., Diamos, G., Miller, J., Peng, K., Ping, W., Raiman, J., & Zhou, Y. (2017). Deep voice 2: Multi-speaker Neural text-to-speech. *Advances in Neural Information Processing Systems*, *30*.

Habib, M., Faris, M., Qaddoura, R., Alomari, M., Alomari, A., & Faris, H. (2021). Toward an automatic quality assessment of voice-based telemedicine consultations: A Deep Learning approach. *Sensors*, *21*(9), 3279.

Helander, E., Schwarz, J., Nurminen, J., Silen, H., & Gabbouj, M. (2008). On the impact of alignment on voice conversion performance. *Interspeech*, 1453–1456.

Jemine, C., et al. (2019). Real-time voice cloning published at Université de Liège, Liège, belgique.

Jia, Y., Zhang, Y., Weiss, R., Wang, Q., Shen, J., Ren, F., Nguyen, P., Pang, R., Lopez Moreno, I., Wu, Y., et al. (2018). Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *Advances in Neural Information Processing Systems*, *31*.

Jiale, Y., & Ying, Z. (2020). Visualization method of sound effect retrieval based on UMAP. *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, *1*, 2216–2220.

Kjartansson, O., Sarin, S., Pipatsrisawat, K., Jansche, M., & Ha, L. (2018). Crowd-sourced speech corpora for Javanese, Sundanese, Sinhala, Nepali, and Bangladeshi Bengali. *SLTU*, 52–55.

Mean Opinion Score (MOS) revisited: Methods and applications, limitations and alternatives. (n.d.).

Mei, X., Huang, Q., Liu, X., Chen, G., Wu, J., Wu, Y., Zhao, J., Li, S., Ko, T., Tang, H. L., et al. (2021). An encoder-decoder based audio captioning system with transfer and reinforcement learning. *arXiv preprint arXiv:2108.02752*.

Neekhara, P., Hussain, S., Dubnov, S., Koushanfar, F., & McAuley, J. (2021). Expressive neural voice cloning. *Asian Conference on Machine Learning*, 252–267.

Rix, A. W., Beerends, J. G., Hollier, M. P., & Hekstra, A. P. (2001). Perceptual Evaluation of Speech Quality (PESQ)- A new method for speech quality assessment of telephone networks and codecs. *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, *2*, 749–752.

Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerrv-Ryan, R., et al. (2018). Natural TTS synthesis by conditioning wavenet on Mel- Spectrogram predictions. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4779–4783.

Sodimana, K., De Silva, P., Sarin, S., Kjartansson, O., Jansche, M., Pipatsrisawat, K., & Ha, L. (2018). A step-by-step process for building tts voices using open source data and frameworks for Bangla, Javanese, Khmer, Nepali, Sinhala, and Sundanese. *SLTU*, 66–70.

Van Den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K., et al. (2016). Wavenet: A Generative model for raw audio. *arXiv preprint arXiv:1609.03499*, *12*.

Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., et al. (2017). Tacotron: Towards end-to-endspeech synthesis. *arXiv preprint arXiv:1703.10135*.

Zhang, H., & Lin, Y. (2022). Improve few-shot voice cloning using Multi-modal learning. *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8317–8321.