TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

A
PROJECT PROGRESS REPORT
ON

DESIGN AND IMPLEMENTATION OF A NEPALI TEXT-TO-SPEECH SYSTEM

SUBMITTED BY:
ASHUTOSH BHATTARAI (PUL078BEI007)
APALA TIMALSINA (PUL078BEI008)

SUBMITTED TO:
DEPARTMENT OF ELECTRONICS & COMPUTER ENGINEERING

October 10, 2025

Acknowledgments

We sincerely thank our supervisor, Jyoti Tandulkar Sir, for his valuable guidance, advice, and support in shaping the idea of this project and assisting us throughout its initial stages.

We also extend our gratitude to all the authors and contributors whose work we have referenced. The development of Nepali TTS has been informed by numerous models and research at various scales, and as we progress further, we carry forward the knowledge and insights gained from their efforts.

In particular, we acknowledge the team behind the OpenSLR SLR43 dataset (K. Sodimana et al., 2018) [1], which has provided high-quality Nepali speech recordings essential for this project. Their contributions to open-source TTS resources have significantly aided our work and learning.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
ASR	Automatic Speech Recognition
CNN	Convolutional Neural Network
DNN	Deep Neural Network
DSP	Digital Signal Processing
GAN	Generative Adversarial Network
G2P	Grapheme-to-Phoneme
GPU	Graphics Processing Unit
JSON	JavaScript Object Notation
LSTM	Long Short-Term Memory
ML	Machine Learning
MFCC	Mel Frequency Cepstral Coefficients
Mel	Mel-Spectrogram
NFC	Nepali Font Converter
NLP	Natural Language Processing
OCR	Optical Character Recognition
OpenSLR	Open Speech and Language Resources
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
SLR	Speech and Language Resources
TTS	Text-to-Speech
WAV	Waveform Audio File Format
WER	Word Error Rate
SR	Sampling Rate
HiFi-GAN	High-Fidelity Generative Adversarial Network
MOS	Mean Opinion Score

1. Introduction

1.1 Background

Our project focuses on the development of a localized Nepali Text-to-Speech (TTS) system to facilitate the comprehension of textual documents, making them accessible to Nepali individuals who may face challenges in reading or writing Nepali but can understand and speak the language. Nepali, being a low-resource language, has seen limited progress in the development of high-quality Nepali speech synthesis tools and TTS systems. There is a need for the development and deployment of Linguistic tools and Assistive tools in the native language to provide inclusivity. Our project fits into this context where we use modern deep learning architectures like FastSpeech 2 model for mel-spectrogram generation and HiFi-GAN vocoders for natural Nepali speech generation. The system aims to deliver high-quality, natural, and intelligible synthesized speech output.

Additionally, the project evaluates the synthesized speech using objective and subjective evaluation metrics such as Mean Opinion Score (MOS), Word Error Rate (WER), and other intelligibility measures to assess the quality and naturalness of generated speech.

1.2 Problem statements

A considerable portion of Nepal's population—including elderly individuals, people with visual impairments, residents of rural regions, and those with limited literacy—often encounter printed or digital Nepali text such as government notices, public documents, or educational materials but find it difficult to comprehend them independently.

To bridge this linguistic and accessibility gap, our project proposes an assistive technology capable of converting Nepali text into natural and intelligible speech. This promotes digital accessibility, inclusive communication, and equitable access to information for all.

1.3 Objectives

- To develop a Nepali Text-to-Speech system that generates natural, intelligible, and expressive speech from Nepali text, starting with a single-speaker model as the baseline for further multi-speaker extensions.
- To evaluate the system's performance in terms of intelligibility, naturalness, and pronunciation accuracy.

1.4 Scope

- The system primarily focuses on standard printed Nepali text. It does not support mixed-language content, colloquial speech, or handwritten text.
- It will generate audio outputs from digital and scanned documents but will not perform real-time speech recognition or translation.
- Extended features may include simplifying complex terminologies and technical content in documents to make them more comprehensible for general audiences.

2. Literature Review

2.1 Related Works

Several research efforts have contributed to the development of Nepali Text-to-Speech (TTS) and Optical Character Recognition (OCR) systems in recent years. **Shruti: A Nepali Book Reader** [2]which uses Nepali Text-to-Speech Synthesis using Tacotron2 for Melspectrogram Generationis , is an AI-based application that converts Nepali PDF books into human-like audiobooks. It employs PyTesseract OCR to extract text, including complex non-Unicode fonts, and uses a modified Tacotron 2 model with a HiFi-GAN vocoder to synthesize high-quality audio. With a Mean Opinion Score (MOS) of 4.04, Shruti significantly improved accessibility to Nepali literature for visually impaired and reading-challenged individuals. **Bajracharya et al. (2018)** [3] developed one of the earliest Nepali TTS systems using a concatenative speech synthesis approach based on unit selection. Integrated with the NVDA screen reader, it produced intelligible speech but suffered from overlapping and echoing artifacts caused by phoneme misalignments. Despite these issues, it served as a foundation for future Nepali TTS research. **Basnet (2021)** [4] introduced the use of the autoregressive WaveNet model for Nepali speech synthesis, trained on the SLR43 and SLR54 datasets. This work achieved better intelligibility and fluency than concatenative systems but faced limitations in prosody and accent due to speaker variability and data quality. A follow-up study by **Basnet et al. (2021)** []employed a custom dataset, though performance remained constrained by noisy recordings and limited training iterations. **Kumar et al. (2023)** [5] later conducted a comparative evaluation of neural TTS architectures across thirteen Indian languages, including Nepali, and concluded that pairing FastPitch (Łańcucki, 2021) with HiFi-GAN (Kong et al., 2020) produced the most natural and efficient synthesized speech. This study emphasized the importance of combining non-autoregressive models with high-fidelity vocoders for low-resource languages. More recently, a **Nepali TTS study (2023)** [6] implemented the FastPitch–HiFi-GAN combination using the OpenSLR and ILPRL-NAB datasets, achieving MOS scores of 3.70 and 3.40 respectively—demonstrating notable progress in naturalness and speech quality. In parallel, OCR development for Nepali has also advanced. A initiative titled **"Nepali OCR"** - A Major Project of seniors from 2075 Batch designed a system for recognizing handwritten Nepali text using a Convolutional Recurrent Neural Network (CRNN) that combines CNN-based feature extraction with RNN-based sequence recognition. Trained on a dataset of approximately 80,000 handwritten Nepali images, the system successfully converted printed and handwritten text into editable digital formats, marking an important step toward robust Nepali OCR solutions.

2.2 Related theory

2.3 Overview of TTS Evolution

The field of Text-to-Speech (TTS) synthesis has evolved remarkably over the past two centuries. Early attempts in the late 18th century involved mechanical speaking machines that imitated human vocal organs. Significant progress began in the mid-20th century, when Noriko Umeda and colleagues developed one of the first fully functional TTS systems for English. While intelligible, the speech lacked fluency and naturalness.

Subsequent developments, such as concatenative synthesis (joining pre-recorded segments) and parametric synthesis using Hidden Markov Models (HMMs), improved the smoothness of generated speech. Deep learning ushered in a new era:WaveNet (2016) produced high-fidelity waveform-level audio; Tacotron and Tacotron 2 combined sequence-to-sequence models with attention mechanisms to improve prosody and expressiveness; and FastSpeech / FastSpeech 2 introduced non-autoregressive architectures for faster, more stable synthesis without compromising naturalness. Modern neural vocoders like WaveGlow and HiFi-GAN enable near-human-quality speech synthesis, extending these capabilities to low-resource languages such as Nepali.

2.4 Integration of OCR and TTS Systems

Optical Character Recognition (OCR) and TTS are complementary technologies in machine reading systems. OCR converts printed or handwritten text images into machine-readable text, while TTS converts this text into audible speech. Integrating the two allows seamless reading of printed or scanned documents, improving accessibility for visually impaired users and individuals with reading difficulties. This modular design also supports future upgrades—such as replacing a pre-trained OCR engine with a Transformer-based model or updating the TTS module for higher naturalness.

Our project specifically integrates OCR and TTS to read printed Nepali documents aloud, allowing interoperability between models and enabling future improvements without redesigning the entire pipeline.

2.5 Optical Character Recognition (OCR)

OCR is the process of converting text in images to digital, machine-readable form. Nepali OCR faces unique challenges due to the complexity of Devanagari script, including ligatures, conjunct consonants, vowel signs, and diacritics.

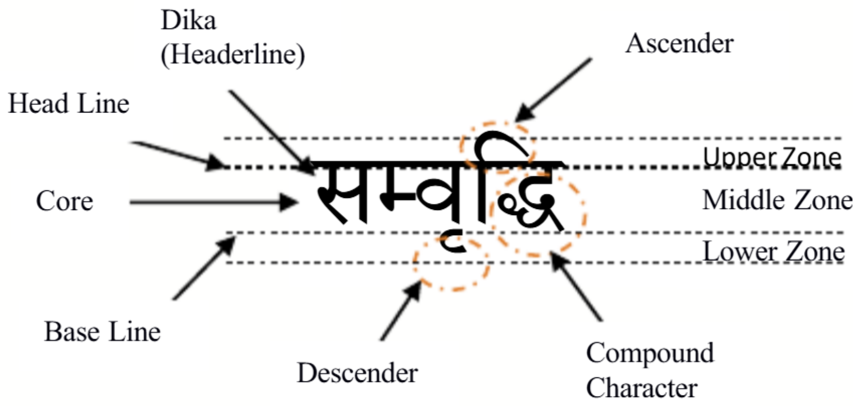

Figure 2.1: Structure of a Nepali Word

2.5.1 OCR Libraries and Algorithms

Common Python libraries include **PyTesseract** and **EasyOCR**. There are two core types of OCR algorithms:

- **Pattern Matching:** Compares images of glyphs to stored templates pixel-by-pixel. Works well for typewritten text with known fonts but fails with unknown fonts.

- **Feature Extraction:** Decomposes glyphs into features such as lines, loops, intersections, and directions. Reference characters are stored as abstract vectors. Most modern OCR and handwriting recognition systems use this method.

2.5.2 Tesseract OCR

Tesseract is an open-source OCR engine known for accuracy, flexibility, and multilingual support. Using LSTM-based sequence modeling and adaptive character segmentation, it supports Devanagari scripts including Nepali. For low-resource scripts, it can be retrained with labeled datasets to improve recognition accuracy. Advanced OCR systems may also employ CRNN or Transformer-based architectures for end-to-end recognition.

2.6 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs) are widely used in image processing tasks due to their ability to detect patterns and spatial relationships. They are particularly effective for feature extraction in OCR and computer vision tasks, including character recognition, object detection, and image segmentation.

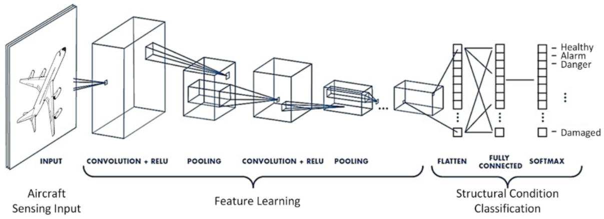

Figure 2.2: CNN Architecture

2.7 Text-to-Speech (TTS) Systems

TTS converts written text into spoken voice and is widely used in accessibility tools, virtual assistants, and audiobooks. Modern TTS systems aim to generate speech that is not only intelligible but also natural and expressive, capturing prosody, intonation, and emotional cues. Advanced models can adapt to different speaking styles, accents, and speaking rates, providing a more human-like listening experience across diverse applications. The process involves:

1. **Text Preprocessing:** Cleaning input text by correcting punctuation, expanding numbers, and standardizing the format.

2. **Acoustic Modeling:** Generating intermediate representations such as mel-spectrograms from text or phonemes.

3. **Vocoding:** Converting the spectrogram into a waveform using a vocoder.

2.7.1 Text Normalization and Preprocessing

After OCR extraction, text normalization ensures consistency for TTS. Key steps include:

- Number expansion- Converting Devanagari numerals to their word equivalents.

- Punctuation handling- Standardizing or removing inconsistent symbols.

- Abbreviation expansion- Expanding commonly used short forms for clarity.

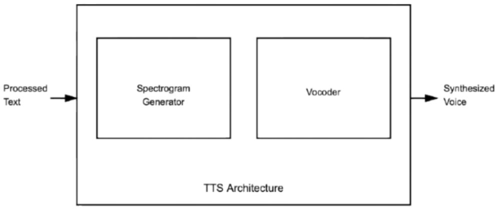

Figure 2.3: TTS Architecture

2.7.2 Grapheme-to-Phoneme (G2P) Conversion

The Grapheme-to-Phoneme (G2P) conversion is a vital intermediary stage in TTS pipelines. It transforms written characters (graphemes) into phonemic representations that correspond to how words are pronounced. In Nepali, which uses the Devanagari script, G2P mapping is complex because pronunciation can vary based on word position, inherent vowels, and schwa deletion rules.

- **Character-based G2P:** Each character is directly tokenized and mapped to its phoneme. This approach is computationally simple but may overlook contextual nuances.
- **Rule-based / Akshara-based G2P:** The text is segmented into aksharas—visual syllables consisting of consonant clusters, vowels, and modifiers - aligning with the phonetic structure of Nepali. Advanced methods use sequence-to-sequence networks or Weighted Finite-State Transducers (WFSTs). This structure aligns more closely with the phonetic nature of Nepali.

2.7.3 Acoustic Modeling and FastSpeech2

Acoustic modeling is the stage where linguistic or phonemic features are transformed into intermediate acoustic representations, typically mel-spectrograms. These representations capture time–frequency information corresponding to human auditory perception. **FastSpeech2** (Ren et al., 2021) is a non-autoregressive TTS model that predicts mel-spectrograms in parallel, ensuring stable and fast synthesis. Its architecture includes:

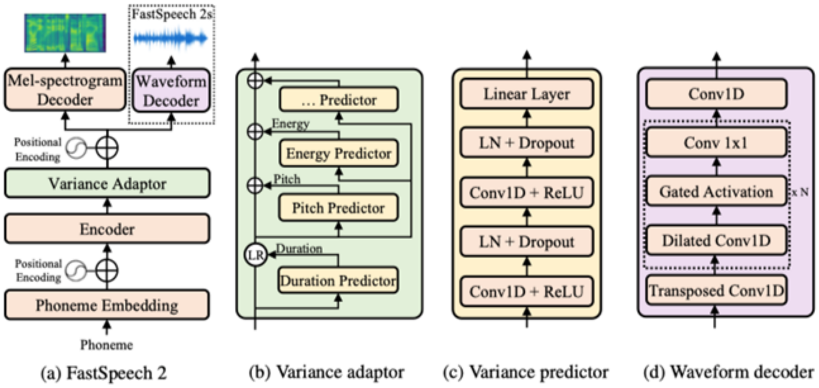

Figure 2.4: FastSpeech2 Model

- **Encoder:** Converts phoneme embeddings to contextual representations.
- **Variance Adaptor:** Predicts duration, pitch, and energy.

- **Decoder:** Generates mel-spectrogram frames.

FastSpeech2 is trained using multiple loss functions, including mel-reconstruction (L1) loss and auxiliary variance prediction losses. This design results in faster training and inference without sacrificing speech naturalness, making it well-suited for low-resource languages like Nepali.

FastSpeech 2 follows a non-autoregressive sequence-to-sequence structure that predicts all Mel-spectrogram frames simultaneously, overcoming the slow inference and instability issues found in traditional autoregressive models like Tacotron 2.

2.8 Mel-Spectrogram

A Mel-Spectrogram is a time–frequency representation of an audio signal, commonly used as an intermediate feature in modern Text-to-Speech (TTS) systems. It provides a visual and numerical representation of how the frequency content of a signal changes over time, but mapped onto the Mel scale, which approximates the way humans perceive pitch. In speech synthesis, a raw waveform contains too much complex information for direct modeling. To simplify this, the waveform is first transformed into a spectrogram using the Short-Time Fourier Transform (STFT), which divides the signal into short overlapping frames and computes the frequency content for each. However, since human auditory perception is not linear — we are more sensitive to differences in lower frequencies than in higher ones — the frequency axis is converted to the Mel scale, which spaces frequencies in a perceptually uniform manner. The resulting Mel-Spectrogram thus captures the energy distribution across perceptually relevant frequency bands over time.

2.8.1 Role in TTS Systems

In TTS pipelines, the Mel-Spectrogram serves as a key intermediate representation between text and the final audio waveform. Modern neural TTS models such as FastSpeech 2, Tacotron 2, Transformer TTS, and Glow-TTS first predict the Mel-Spectrogram from input text or phonemes, capturing essential prosodic features like intonation, rhythm, and stress. Tacotron 2 and Transformer TTS use attention-based architectures to model sequence dependencies, while FastSpeech 2 and Glow-TTS employ non-autoregressive approaches for faster and more stable training. The predicted Mel-Spectrogram is then converted into waveform using vocoders such as WaveGlow, HiFi-GAN, or WaveNet, which produce high-fidelity, natural-sounding speech. This two-stage pipeline—text-to-Mel and Mel-to-waveform—forms the basis of most modern neural TTS systems, including single- and multi-speaker models.

For Nepali TTS, the Mel-Spectrogram helps capture the rich vowel sounds, tonal variations, and syllabic timing characteristic of Nepali speech, ensuring that the synthesized audio retains naturalness and clarity.

2.9 Vocoder: Mel-Spectrogram to Waveform Synthesis

The vocoder module converts Mel-Spectrograms into audible speech waveforms. The evolution of vocoder technologies reflects the broader progress in TTS systems:

- **Traditional approaches:** Methods such as Griffin-Lim relied on iterative signal processing to reconstruct phase information but suffered from low naturalness.

- **Neural vocoders:** Revolutionized the process by directly learning waveform generation from spectrograms.

2.9.1 Examples of Neural Vocoders

- **WaveNet (Van Den Oord et al., 2016):** An autoregressive model capable of producing extremely high-quality speech at the cost of slow generation speed.
- **WaveGlow (Prenger et al., 2019):** A flow-based model that generates speech in parallel, balancing quality and computational efficiency.
- **HiFi-GAN (Kong et al., 2020):** A GAN-based vocoder optimized for real-time synthesis with excellent perceptual quality. It achieves this using multi-period and multi-scale discriminators to capture both global and local acoustic patterns.

Table 2.1: Summary of Neural Vocoders

Vocoder	**Type**	**Inference**	**Quality**
WaveNet	Autoregressive	Slow	Very High
WaveGlow	Flow-based	Fast (Parallel)	High
HiFi-GAN	GAN-based	Real-time	Excellent

Recent studies demonstrate that combining **FastSpeech2** with **HiFi-GAN** achieves state-of-the-art performance in terms of both Mean Opinion Score (MOS) and inference latency, particularly in low-resource linguistic environments.

2.10 Evaluation Metrics for the TTS System

Evaluating a Text-to-Speech (TTS) system requires both objective and subjective metrics to determine how natural, intelligible, and accurate the synthesized speech is compared to real human speech. For the Nepali TTS system, the following metrics are commonly used to assess performance:

- **Mel Cepstral Distortion (MCD):** Mel Cepstral Distortion (MCD) is an objective measure used to quantify the difference between the spectral features of synthesized and natural speech. It evaluates how closely the generated speech waveform's spectral envelope matches that of the reference (ground truth) signal. MCD is calculated based on the Mel-Frequency Cepstral Coefficients (MFCCs), which represent the short-term power spectrum of sound. A lower MCD value indicates that the synthesized speech is more similar to the natural reference, implying better spectral quality. In Nepali TTS evaluation, MCD helps assess how accurately the model captures the tonal and vowel-rich characteristics of the Nepali language.
- **F0 Root Mean Square Error (F0-RMSE):** The F0 Root Mean Square Error (F0-RMSE) measures the difference between the fundamental frequency (pitch) contours of the synthesized and reference speech signals. Since pitch plays a vital role in expressing intonation and prosody, this metric evaluates how well the model reproduces natural pitch variations. A lower F0-RMSE value indicates that the model more accurately captures the pitch pattern and rhythm of the original speaker. This is especially important for Nepali TTS, as proper pitch contouring affects the perceived naturalness and emotional tone of synthesized Nepali speech.
- **Automatic Speech Recognition (ASR)-Based Evaluation:** ASR-based evaluation provides an indirect measure of intelligibility by transcribing the generated speech using an Automatic Speech

Recognition system and comparing it with the original input text. The resulting Word Error Rate (WER) or Character Error Rate (CER) indicates how understandable the synthesized speech is to a machine listener. In Nepali TTS, ASR-based evaluation is useful due to the limited availability of large-scale subjective evaluation datasets. A lower WER or CER implies that the synthesized Nepali speech is clear and easily recognizable, reflecting good pronunciation and intelligibility.

- **Mean Opinion Score (MOS):** The Mean Opinion Score (MOS) is a subjective metric obtained by collecting human listeners' opinions on the quality of synthesized speech. Participants rate samples on a 5-point scale (typically from 1 – "Bad" to 5 – "Excellent") based on naturalness, clarity, and similarity to human voice. MOS provides insights into how real users perceive the TTS system's performance, which objective metrics alone cannot capture. For Nepali speech synthesis, MOS evaluation helps determine whether the voice sounds natural, pleasant, and contextually appropriate to native Nepali listeners.

Table 2.2: Evaluation Metrics for Nepali TTS System

Metric	**Type**	**Measures**	**Interpretation**
MCD	Objective	Spectral similarity	Lower = closer to natural speech
F0-RMSE	Objective	Pitch contour accuracy	Lower = better pitch reproduction
WER / CER	Objective	Intelligibility	Lower = higher clarity & recognition
MOS	Subjective	Naturalness, clarity, similarity	Higher = more natural & pleasant

3. Proposed Methodology

3.0.1 Feasibility Study

The feasibility study was conducted to assess the technical, operational, and economic viability of developing an end-to-end Nepali TTS system.

- **Technical Feasibility:** Existing open-source technologies were reviewed to identify suitable tools for each system component. Tesseract OCR was selected for text extraction due to its Devanagari support and adaptability for Nepali. Among TTS models, FastSpeech 2 was chosen for its fast, stable, and expressive speech generation, while HiFi-GAN and WaveGlow were selected as vocoders for high-quality waveform synthesis. Available Nepali datasets from OpenSLR and Mozilla Common Voice were also evaluated for adequacy and coverage.

- **Operational Feasibility:** The system is designed to process standard printed Nepali text and convert it into natural and intelligible speech. It supports both digital and scanned documents, enabling accessibility for users who prefer or require audio-based interaction with written content. However, it does not handle mixed-language inputs, handwritten text, or real-time speech recognition and translation, keeping the operational focus clear and manageable. The proposed system is particularly beneficial for visually impaired individuals, students, and general readers seeking easier access to printed Nepali materials. Additionally, the generated audio can assist in education, digital archiving, and inclusive content dissemination. Future extensions may include integration with mobile accessibility platforms, simplification of technical or complex Nepali texts, and eventual expansion to regional languages.

- **Economic Feasibility:** Economic feasibility evaluates whether the project is financially practical and can be completed within the available resources. Most key components such as datasets, pre-trained models, and frameworks are open-source, reducing development costs. Additional costs may arise if manually curated datasets are required for model fine-tuning or evaluation. Despite minimal expenditures, the long-term benefits of enhanced accessibility for Nepali literature outweigh the costs, making the project economically viable.

3.0.2 Requirement Analysis

Functional Requirements:

- Extract printed Nepali text using OCR.
- Normalize and preprocess text for consistency.
- Generate speech using FastSpeech2 and HiFi-GAN.
- Produce high-quality, natural Nepali speech output.

Non-Functional Requirements:

- Accuracy: Minimize OCR and pronunciation errors.
- Naturalness: Maintain human-like prosody and tone.
- Efficiency: Achieve low-latency synthesis for real-time applications.
- Scalability: Enable future expansion to multi-speaker or multilingual setups.

3.0.3 Data Collection

Publicly available Nepali speech datasets were utilized for this research to ensure high-quality and reproducible results. Two primary sources from the OpenSLR repository were used: SLR43 and SLR143.

- **OpenSLR SLR43:** The "High-Quality Nepali TTS Dataset" (SLR43) comprises recordings from 19 native female speakers, totaling 2,064 utterances (approximately 2.8 hours of audio). Developed by Google in Nepal, the dataset features high-fidelity 48 kHz, 16-bit mono audio recordings aligned with accurately transcribed Nepali text. The speech material includes a balanced mix of literary readings, news articles, and everyday conversational phrases, offering phonetic and prosodic diversity. This dataset has been widely recognized as one of the most reliable resources for Nepali TTS research due to its consistent quality and clean recording conditions.
- **OpenSLR SLR143:** The SLR143 dataset complements SLR43 by including both male and female speakers, containing approximately 1,500 utterances (1.25 hours). Although smaller in scale, it introduces essential gender and voice diversity, which is valuable for training and evaluating multi-speaker or gender-balanced TTS systems. The recordings are well-segmented and transcribed, representing a broader acoustic variety across speakers.

Initially, experiments will be conducted using the **SLR43 dataset** to establish a robust baseline single-speaker model, ensuring consistent alignment between text and speech. This allows for stable mel-spectrogram generation, pronunciation consistency, and model convergence.
Later, both datasets (SLR43 and SLR143) will be merged to develop a multi-speaker Nepali TTS system. The datasets will be unified into a standardized structure by:

- Converting all audio to a uniform format (22.05 kHz, 16-bit, mono);
- Normalizing transcripts to a consistent text format;
- Creating a unified metadata file including speaker identifiers;

- Adding a speaker ID column for embedding-based multi-speaker training.

Table 3.1: Summary of Nepali Speech Datasets Used

Dataset	**Speakers**	**Duration (hrs)**	**Utterances**	**Gender**
SLR43	19	2.8	2,064	Female only
SLR143	Multiple	1.25	1,500	Male & Female

This integration will enable the model to learn speaker-dependent embeddings, capturing unique vocal characteristics while maintaining language consistency. The multi-speaker system will thus provide flexibility for generating both male and female Nepali voices and serve as a foundation for future speaker-adaptive TTS research.

Future Data Expansion: As the TTS system evolves and demands higher quality or stylistic diversity, it is anticipated that the existing public datasets may not be sufficient to achieve desired naturalness and speaker variation. Therefore, additional data sources are being considered to further expand the training corpus. These include:

- **Audiobook Recordings:** Narrated Nepali audiobooks provide expressive and well-articulated speech, ideal for improving prosody modeling and intonation control.They may be incorporated from the available recordings found on YouTube.

- **News Broadcast Recordings:** Speech segments from televised news broadcasts (e.g., *Kantipur TV*, *Nepal TV*), sourced from publicly available YouTube archives, offer formal and clear pronunciation from professional anchors.

These additional sources will be processed using the same normalization, segmentation, and metadata alignment pipeline, ensuring they seamlessly integrate into the joint dataset. This approach supports the creation of a scalable and adaptable Nepali TTS system capable of synthesizing speech across different speaking styles, domains, and speaker identities.

3.0.4 Data Preprocessing

The preprocessing pipeline prepares textual and audio data for model training. The goal is to ensure data uniformity, linguistic consistency, and high-quality audio-text alignment, which is critical for robust TTS performance.

OCR-based Text Extraction

Scanned Nepali textbooks, newspapers, and other documents were processed using Tesseract OCR with full Devanagari support. Outputs were stored in `.json` and `.txt` formats.

Text Normalization and Preprocessing

Text normalization and preprocessing were carried out to prepare consistent and model-friendly transcripts. Key steps included:

1. **Unicode Normalization:** All characters were converted to NFC form to maintain consistent encoding and also for debugging, reproducibility, and clean preprocessing .

2. **Number, Date, and Currency Expansion:** Numerals, years, and dates were converted to Nepali words using a Python script.

3. **Removal of Unwanted Characters:** Non-standard symbols, foreign characters, and unseen Unicode symbols were removed to avoid pronunciation errors.

4. **Sentence Segmentation:** Lengthy sentences were broken into smaller units to improve TTS alignment and model stability.

5. **Stop Token Addition:** Manual addition of appropriate stop tokens at the end of each sentence ensures that the model learns proper phrase termination.

6. **Abbreviation Expansion:** Common abbreviations were expanded for clarity.

Audio Preprocessing

Audio processing was conducted to ensure uniformity and reduce artifacts that can degrade model performance. Steps included:

- Resampling all audio to **22.05 kHz, 16-bit, mono** format.
- Trimming silence at the beginning and end of recordings.
- Segmenting long recordings into 2–12 second clips for optimal model training.
- Filtering out noisy or low-quality recordings based on energy and signal-to-noise ratio.

Feature Extraction

Mel-spectrograms were generated using Librosa with standardized parameters for all datasets. A manifest file was created for each dataset, containing:

- File path of each audio sample.
- Preprocessed transcript.
- Phoneme durations and alignment information.

Precomputation of features was performed to accelerate training and reduce runtime overhead.

3.0.5 Proposed System Design

The proposed Nepali TTS system is a modular pipeline integrating OCR, text preprocessing, and TTS modules. The system is designed to be extensible for multi-speaker and domain-adaptive speech synthesis.

OCR Module

The OCR module extracts text from printed Nepali documents using Tesseract OCR. Outputs are stored in structured `.json` and `.txt` files. Manual correction addresses OCR errors, while future enhancements may include Transformer-based or CRNN OCR architectures to further improve accuracy.

Text Preprocessing Module

This module ensures linguistic and syntactic consistency, and prepares text suitable for TTS. Specific tasks include:

- Unicode normalization and punctuation standardization.
- Expansion of numerals, dates, years, and currency values into Nepali words.
- Sentence segmentation and stop token addition to aid alignment and prosody.
- Abbreviation expansion and optional transliteration for multilingual support.
- Handling of special characters and removal of unseen or non-standard symbols.

TTS Module

The TTS module converts preprocessed text into natural-sounding speech. It is composed of two main components: acoustic modeling and waveform synthesis. The module is designed to handle single-speaker as well as future multi-speaker synthesis.

1. Acoustic Modeling Acoustic modeling is performed using **FastSpeech2**, a non-autoregressive model that generates mel-spectrograms directly from input text sequences. Key features and processes include:

- **Variance Adaptor:** FastSpeech2 incorporates a variance adaptor that predicts the *duration, pitch, and energy* for each segment. This allows the model to generate more expressive and natural prosody, avoiding the monotony typical of early TTS models.
- **Text-to-Spectrogram Conversion:** Preprocessed text sequences are mapped to mel-spectrogram representations. This intermediate representation captures the spectral and temporal characteristics of speech, enabling the model to learn patterns such as intonation, stress, and rhythm.
- **Training Considerations:** The model is trained using L1 loss on predicted versus target mel-spectrograms, along with auxiliary losses for pitch and energy. Teacher-forcing is employed during early epochs for faster convergence.
- **Scalability:** The architecture is designed to accommodate additional speaker embeddings, allowing the system to learn unique voice characteristics for multi-speaker synthesis.

2. Waveform Synthesis The waveform synthesis step converts the predicted mel-spectrograms into audio waveforms. HiFi-GAN is used as the vocoder for high-quality, real-time audio generation. Details include:

- **Generative Adversarial Training:** HiFi-GAN uses a generator-discriminator setup to produce waveforms that closely mimic natural speech. Multi-scale and multi-period discriminators ensure both local and global audio fidelity.
- **High-Fidelity Audio:** The vocoder generates 22.05 kHz, 16-bit mono audio, preserving clarity and naturalness across different speaking styles and recording conditions.
- **Speaker Adaptation:** By incorporating speaker embeddings, HiFi-GAN can generate speech for multiple speakers, capturing differences in pitch, timbre, and speaking style.

- **Future Improvements:** Potential enhancements include multi-style modeling, prosody transfer, and fine-tuning on domain-specific corpora (e.g., news broadcasts or audiobooks) to further improve naturalness and expressiveness.

3.0.6 Testing and Evaluation

The model's performance is evaluated using:

Objective Metrics:

- Mel-Cepstral Distortion (MCD): Measures spectral distance between generated and reference speech.
- F0 Root Mean Square Error (F0-RMSE): Evaluates pitch prediction accuracy.
- ASR Word Error Rate (WER): Assesses intelligibility of synthesized speech.

Subjective Metrics:

- Mean Opinion Score (MOS): Human listeners rate speech quality (1–5 scale).
- ABX Preference Test: Compares listener preference between different model outputs.

4. Timeline

4.0.1 Project Timeline and Progress: Nepali TTS System

The project timeline outlines the tasks, target dates, and current progress for the development of the Nepali TTS system. The timeline is structured to reflect both the research and implementation stages.

Task	**Target Week/Date**	**Status**
General literature review, dataset exploration, and foundational learning of TTS models including Tacotron, Tacotron2, FastSpeech2, and vocoders.	Initial Phase	Completed
Testing OCR model: collection of sample text data and evaluation of Tesseract OCR performance.	Sep 19, 2025	Completed
Normalization of OCR output: handling noise, digit expansion, abbreviation expansion, and sentence segmentation. Evaluation of potential preprocessing challenges and preparation for TTS training.	Sep 26 – Oct 3, 2025	Completed
Data preprocessing of audio and textual datasets from OpenSLR, including cleaning, normalization, and feature extraction. Documentation of preprocessing pipeline. G2P methods were tested but difficulties led to proceeding with direct text-to-mel-spectrogram mapping.	Oct 3 – Oct 10, 2025	Completed
Text-to-mel-spectrogram conversion via FastSpeech2 and initial model training. Estimated training duration: 7–8 days.	Oct 10 – Oct 19, 2025	Pending
Vocoder implementation for waveform synthesis (Mel → Audio) using HiFi-GAN [and WaveGlow if possible].	Oct 19 – Oct 24, 2025	Pending
Evaluation of vocoder performance, including objective and subjective metrics for naturalness and intelligibility.	Oct 24 – Oct 31, 2025	Pending
Exam week.	Oct 31 – Nov 14, 2025	Scheduled
Multi-speaker TTS implementation: integration of SLR143 with SLR43, and additional data from news broadcasts (YouTube). Preparation for training with multiple speakers.	Nov 14 – Nov 28, 2025	Planned
Retraining FastSpeech2 for multi-speaker mel-spectrogram generation. Estimated duration: 1 week.	Nov 28 – Dec 5, 2025	Planned

Task	**Target Week/Date**	**Status**
Vocoder re-implementation for multi-speaker audio synthesis (Mel $\rightarrow$ Waveform).	Dec 5 – Dec 9, 2025	Planned
Frontend and backend integration for TTS system demonstration, including web interface and API connectivity. Refining Documentation	December Onwards 2025	Planned

References

[1] Keshan Sodimana, Knot Pipatsrisawat, Linne Ha, Martin Jansche, Oddur Kjartansson, Pasindu De Silva, and Supheakmungkol Sarin. A Step-by-Step Process for Building TTS Voices Using Open Source Data and Framework for Bangla, Javanese, Khmer, Nepali, Sinhala, and Sundanese. In *Proc. The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU)*, pages 66–70, Gurugram, India, August 2018.

[2] Supriya Khadka, Ranju G.C., Prabin Paudel, Rahul Shah, and Basanta Joshi. Nepali text-to-speech synthesis using tacotron2 for melspectrogram generation. pages 73–77, 08 2023.

[3] R. Bajracharya, S. Regmi, B. K. Bal, and B. Prasain. Building a natural sounding text-to-speech system for the nepali language — research and development challenges and solutions. In *Proceedings of the 6th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU 2018)*, pages 152–156, 2018.

[4] Ashok Basnet. Attention and wave net vocoder based nepali text-to-speech synthesis. https://elibrary.tucl.edu.np/handle/123456789/7668, 2021.

[5] G. K. Kumar, P. S. V, P. Kumar, M. M. Khapra, and K. Nandakumar. Towards building text-to-speech systems for the next billion users. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023)*, 2023.

[6] I. Dongol and B. K. Bal. Transformer-based nepali text-to-speech. In *Proceedings of the 20th International Conference on Natural Language Processing (ICON 2023)*, pages 651–656, Goa, India, December 2023.