RSET
RAJAGIRI SCHOOL OF
ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

Project Report On

Automated Video Translator

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Technology

By

Tharasankar S(U2103206)
Neethu Anil Jacob (U2103152)
Shawn Antony Sobi (U2103195)
Vineet Abraham Koshy (U2103214)

Under the guidance of
Ms. Sangeetha Jamal
Assistant Professor

Computer Science and Technology
Rajagiri School of Engineering & Technology (Autonomous)
(Parent University: APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039

April 2025

CERTIFICATE

This is to certify that the project report entitled ***"Automated Video Translator"*** *is a bonafide record of the work done by* ***Tharasankar S(U2103206), Neethu Anil Jacob(U2103152), Shawn Antony Sobi(U2103195), Vineet Abraham Koshy(U2103214)****, submitted to the Rajagiri School of Engineering & Technology (RSET) (Autonomous) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Technology during the academic year 2024-2025.*

Ms. Sangeetha Jamal	**Ms. Sangeetha Jamal**
Project Guide	Project Coordinator
Assistant Professor	Assistant Professor
Dept. of CSE	Dept. of CSE
RSET	RSET

Dr. Preetha K G
Professor & HoD
Dept. of CSE
RSET

ACKNOWLEDGMENT

We wish to express our sincere gratitude towards **Fr. Dr. Jaison Paul Mulerikkal CMI**, Principal, RSET, and **Dr. Preetha K G**, Professor & HoD, Department of Computer Science and Engineering for providing us with the opportunity to undertake our project, ”Automated Video Translator”.

We are highly indebted to our project coordinator, **Ms. Sangeetha Jamal**, Assistant Professor, Dept of CSE, for her valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide **Ms. Sangeetha Jamal**, Assistant Professor, Dept of CSE, for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, We would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Tharasankar S
Neethu Anil Jacob
Shawn Antony Sobi
Vineet Abraham Koshy

Abstract

With the rise of online streaming platforms, the demand for diverse content has grown significantly. Videos in regional languages like Malayalam are gaining popularity. However, language barriers often limit their global reach. Despite the demand, high-quality automated translation and dubbing tools specifically for converting Malayalam videos into English remain scarce. The manual process of recording and translating Malayalam audio into English is time-consuming and requires substantial effort to ensure accuracy in both translation and synchronization.

To address these challenges, this project aims to develop an automated system that efficiently transcribes and translates Malayalam audio into English.

System Features

- Accurate transcription of spoken Malayalam dialogue
- Automatic translation of transcribed text into English
- Voice synthesis to convert translated text into natural-sounding English audio
- Timing adjustment to match the original video dialogue

By leveraging advanced speech recognition, translation, and text-to-speech technologies, this NLP-based software will streamline the translation process, reduce manual effort, and enhance the accessibility of Malayalam content to a global audience.

Contents

Acknowledgment i

Abstract ii

List of Abbreviations vii

List of Figures viii

List of Tables ix

1 Introduction 1

1.1 Background 1

1.2 Problem Definition 2

1.3 Scope and Motivation 2

1.4 Objectives 2

1.5 Challenges 3

1.6 Assumptions 3

1.7 Societal / Industrial Relevance 3

1.8 Organization of the Report 4

1.9 Chapter Conclusion 4

2 Literature Survey 6

2.1 Listen Attentively, and Spell Once: Whole Sentence Generation via a Non-Autoregressive Architecture for Low-Latency Speech Recognition(Y. Bai et al.)[1] 6

2.1.1 Introduction and Motivation 6

2.1.2 LASO Model Architecture 6

2.2 Future Cost Modeling in Neural Machine Translation(Duan et al.)[2] 10

2.2.1 Introduction 10

2.2.2 Key Contributions 10
2.2.3 Methodology 10
2.2.4 Results 13
2.2.5 Conclusion 14
2.3 Audio Visual-Automatic Speech Recognition(D. Serdyuk et al.)[3] 14
2.3.1 AV-ASR System Architecture 14
2.3.2 Model Components 15
2.3.3 Methods 15
2.3.4 Video Transformer Front End 16
2.3.5 Audio-Visual Automatic Speech Recognition 18
2.3.6 Conclusion 19
2.4 Arabic Speech Synthesis using Deep Neural Networks(A. Ali et al.)[4] . . . 19
2.4.1 Introduction 19
2.4.2 Deep Neural Networks in Arabic TTS: Enhancing Naturalness and Quality 20
2.4.3 Methodology 21
2.5 Summary and Gaps Identified 29
2.6 Conclusion 29

3 Requirements **31**
3.1 Hardware and Software Requirements 31

4 System Architecture **32**
4.1 Architecture Design 32
4.2 Component Design 33
4.2.1 Audio Extraction Module 33
4.2.2 Audio Preprocessing Module 33
4.2.3 Gender Recognition Module 33
4.2.4 Speech-to-Text Conversion Module 34
4.2.5 Translation Module 34
4.2.6 Text-to-Speech Module 34
4.2.7 Audio Synchronization Module 35
4.2.8 Audio-Video Merging Module 35

4.3 Use Case Diagram 36
4.4 Sequence Diagram 37
4.5 Gantt Chart 38

5 System Implementation 39
5.1 Proposed Methodology 39
5.1.1 Audio Extraction 39
5.2 Audio Preprocessing 39
5.2.1 Gender Recognition 40
5.2.2 Speech-to-Text Conversion 40
5.2.3 Translation 40
5.2.4 Text-to-Speech 41
5.2.5 Audio Synchronization 41
5.2.6 Audio-Video Merging 41
5.3 User Interface Design 41
5.4 Implementation Strategies 42
5.4.1 Audio Extraction 42
5.4.2 Audio Preprocessing 42
5.4.3 Gender Recognition 44
5.4.4 Speech-to-Text Conversion 45
5.4.5 Translation 46
5.4.6 Text-to-Speech 46
5.4.7 Audio Synchronization 47
5.4.8 Audio-Video Merging 47
5.5 Chapter Conclusion 48

6 Results and Discussions 49
6.1 Testing 49
6.1.1 Audio Extraction 49
6.1.2 Audio Separation 50
6.1.3 Audio Segmentation 50
6.1.4 Malayalam Audio to English Audio Conversion 50
6.1.5 Final Output 51

6.2 Quantitative Results 52

6.2.1 Model 52

6.2.2 Execution 53

6.2.3 Discussion 54

6.3 Chapter Conclusion 54

7 Conclusion and Future Scope 55

References 56

Appendix A: Presentation 58

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes 77

Appendix C: CO-PO-PSO Mapping 81

List of Abbreviations

ASR - Automatic Speech Recognition

AV-ASR - Audio-Visual Automatic Speech Recognition

BLEU - Bilingual Evaluation Understudy (translation quality metric)

CNN - Convolutional Neural Network

DNN - Deep Neural Network

LSTM - Long Short-Term Memory

MOS - Mean Opinion Score

NLP - Natural Language Processing

NMT - Neural Machine Translation

RNN - Recurrent Neural Network

TTS - Text-to-Speech

WER - Word Error Rate

List of Figures

2.1 Architecture of the LASO Model 7

2.2 Architecture of Model I, showing the additional loss term for future cost during training. 12

2.3 Architecture of Model II, illustrating the integration of future cost into decoding. 13

2.4 Audio-visual encoder combines audio and visual cues for transcription. . . 14

2.5 Transformer model processes video patches to capture spatial-temporal information. 18

2.6 Process flowchart 20

2.7 Deep Learning Network 24

2.8 Recurrent Neural Networks 25

2.9 Long Short-Term Memory networks 25

2.10 Tacotron model architecture 26

2.11 Tacotron 2 model architecture 26

2.12 Mean Opinion Score of 50 participants 28

4.1 Architectural Diagram 32

4.2 Sequence Diagram 37

4.3 Gantt Chart 38

6.1 Audio is extracted from the input video file and saved as out.mp3 49

6.2 Audio was separated into vocals and accompaniment 50

6.3 Vocals were Segmented and stored as small dialogue segments 50

6.4 Steps involved in converting Malayalam audio to English audio. 51

6.5 Final Output video was generated. 52

6.6 Training and Validation Accuracy and Loss Graph. 52

6.7 Confusion matrix based on validation. 53

List of Tables

2.1 Performance comparison of baseline and proposed models on BLEU scores 13

2.2 Summary of Literature Reviewed 29

6.1 Movie Scene Durations and Execution Times 53

Chapter 1

Introduction

This chapter provides a structured overview of the Automated Video Translator project. It begins with an outline of the project's background, covering current practices and the significance of developing an automated translation tool specifically for Malayalam videos. Following this, the problem definition precisely states the project's goal. Sections on the scope and motivation offer insight into the intended impact and driving factors behind the project, while objectives highlight the key functionalities to be achieved. Challenges and assumptions give an idea of potential limitations and conditions for successful implementation. The chapter further discusses the societal and industrial relevance of the tool and closes with an outline of the report's organization, detailing how the remaining sections unfold.

1.1 Background

With the rise of digital media, there is a growing demand for regional video content to reach audiences beyond linguistic boundaries. Malayalam, a language spoken by millions primarily in the Indian state of Kerala, has a rich collection of cultural and informational video content. However, most Malayalam videos are inaccessible to non-Malayalam speakers, limiting their global reach. Traditional methods of translating and subtitling such videos require substantial manual effort, involving time-intensive tasks like transcription, translation, and synchronization.

The need for an automated translation tool arises from these challenges. Leveraging advancements in speech recognition, natural language processing, and machine translation, an automated system can streamline the translation of Malayalam videos into English. By converting audio to text, translating it, and then re-synthesizing it in English, this project aims to minimize manual work while maintaining translation accuracy and synchronization with the original video. Such a tool not only makes Malayalam content accessible to a broader audience but also supports cross-cultural communication, helping Malayalam creators reach a global platform.

1.2 Problem Definition

To develop an automated system that translates Malayalam dialogues into English, streamlining video content translation and reducing manual effort while ensuring contextual accuracy.

1.3 Scope and Motivation

The scope of the Automated Video Translator project encompasses creating a system capable of converting Malayalam audio in videos to English audio, covering the entire process from audio extraction to final video output. In order to guarantee that the translated English audio matches the original timing and context of the film, the system will automatically recognize speech, translate text, and synthesize text to speech.

The main motivation behind this project is to bridge the language gap and promote accessibility for Malayalam content to a global audience. Creating a tool to automate the translation of Malayalam videos into English will help to address this gap, allowing non-Malayalam speakers to engage with and appreciate regional content.Another driving factor is the high demand for efficient, cost-effective translation solutions. Current manual translation methods are time-consuming and require significant resources, often making them impractical for widespread sharing. Therefore by automating the translation process, this project aims to offer a scalable solution.

1.4 Objectives

1. Develop an automated system to translate Malayalam audio in videos to English audio, ensuring accurate speech-to-text conversion.

2. Implement a reliable translation model that preserves the context and meaning of the original Malayalam content.

3. Integrate text-to-speech synthesis to generate natural-sounding English audio that is synchronized with the original video.

1.5 Challenges

1. Speech Recognition Accuracy: Ensuring the translated output conveys the intended meaning of the original Malayalam audio.
2. Translation Quality: Ensuring accurate translation of complex sentences, cultural nuances, and context.
3. Real-time Processing Limitations: Delays in translation could impact synchronization between video and audio.
4. Data Availability: Limited datasets for Malayalam audio, which may affect training of speech-to text and translation models.
5. Focus is only on Malayalam videos

1.6 Assumptions

1. Clear Audio: Input videos have distinguishable audio with minimal background noise.
2. Moderate Speech: Speech patterns are clear, with moderate speed for optimal transcription.
3. Accurate Timestamps: Speech start and stop times are correctly detected.

1.7 Societal / Industrial Relevance

This project has significant industrial relevance in sectors such as media, entertainment, education, and digital content creation. In the media industry, automated translation solutions can help regional content producers reach international markets by providing accurate and accessible translations. This tool can also serve educational platforms looking to share local knowledge globally, enabling multilingual learning resources. By streamlining the translation process, this project supports cross-cultural communication and expands market reach for businesses aiming to cater to a diverse, multilingual audience.

1.8 Organization of the Report

The structure of the report is as follows: The study's background, problem definition, scope and motivation, objectives, obstacles, assumptions, social and industrial relevance, and report organization are all covered in the introduction, which is given in Chapter 1. The literature review is the main topic of Chapter 2, which covers a variety of current approaches, their methodologies, findings, and identified research gaps. It identifies the shortcomings that the project seeks to fill and ends with a summary of the reviewed efforts. Chapter 3 focuses on the hardware and software requirements for building this project. The overall system design and architecture is explained in detail in chapter 4. It also consists of use case diagram as well as sequence diagram to understand the workflow better. The implementation of the entire system is explained in detailed in chapter 5. This chapter includes the explanation of each module in depth and how each module is implemented. Chapter 6 includes the results of the implementation done. It also contains the accuracy results regarding training model for gender identification. Also a few observations were made regarding the cases where the system does not behave as intended. Finally chapter 7 gives an overview about the project and also a few extensions are proposed to enhance the applicability of the system.

1.9 Chapter Conclusion

This chapter provided a foundational overview of the Automated Video Translator project, covering its background, problem definition, and objectives. We discussed the current challenges in translating Malayalam video content for a broader audience and outlined the scope and motivation behind creating a tool that automates this process. The objectives aim to ensure high accuracy in speech recognition, contextual translation, and seamless audio synchronization, while also being cost-effective and accessible to users across various industries. Key challenges, such as achieving synchronization and handling dialectal variations, were also highlighted, emphasizing the technical complexity of the project. Additionally, the chapter addressed the industrial relevance of this tool in media, education, and digital content, setting the stage for a solution that enables broader accessibility and cross-cultural communication.

Chapter 2

Literature Survey

The literature survey presented in this chapter explores foundational research and technologies that support the development of the Automated Video Translator. By reviewing four key papers, we gain insights into advancements in speech recognition, machine translation, and text-to-speech synthesis, each contributing essential components for our project.

2.1 Listen Attentively, and Spell Once: Whole Sentence Generation via a Non-Autoregressive Architecture for Low-Latency Speech Recognition(Y. Bai et al.)[1]

2.1.1 Introduction and Motivation

Recent advancements in ASR have leveraged sequence-to-sequence models with attention mechanisms, achieving impressive accuracy. However, the inherent sequential nature of autoregressive models results in increased latency, limiting their use in scenarios where real-time processing is essential. LASO aims to overcome this by using a non-autoregressive framework, where tokens are generated independently, allowing for efficient parallel computation. This approach allows the entire sentence to be generated simultaneously rather than word by word, as in autoregressive models like LAS (Listen, Attend, and Spell)[5]. The key innovation of LASO is its one-pass forward propagation, which eliminates the need for beam search and drastically cuts down inference time.

2.1.2 LASO Model Architecture

The Decoder, Position Dependent Summarizer, and Encoder are the three primary parts of the architecture.

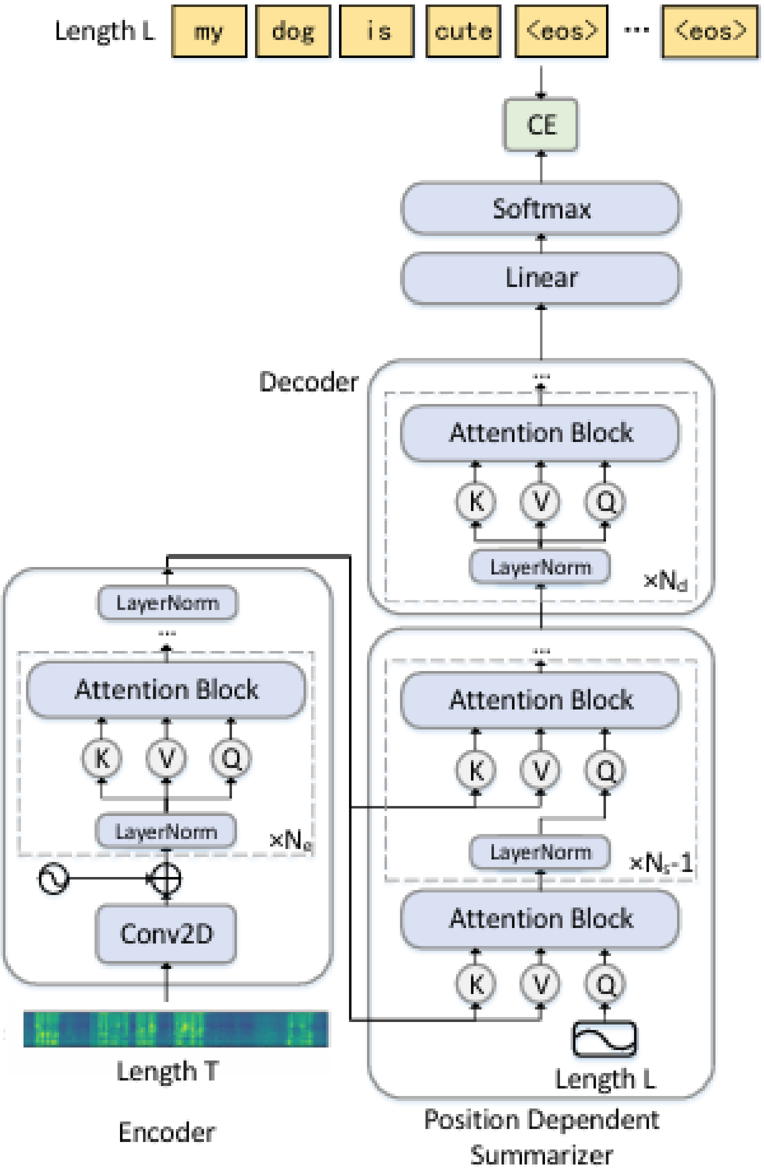


Figure 2.1: Architecture of the LASO Model

Encoder

The role of the encoder is to convert the input acoustic feature sequence x into high-level representations. The encoder utilizes a combination of convolutional neural networks (CNN) and attention mechanisms. Let $\mathbf{x} = [x_1, x_2, \ldots, x_T]$ be the input acoustic features of length T.

- **CNN Subsampling**: The input acoustic features are first processed through a two-layer convolutional network with a stride of 2. This reduces the length of the feature sequence, effectively compressing the input by a factor of 4, which speeds up subsequent processing.

- **Self-Attention Mechanism**: After subsampling, the sequence is passed through several attention blocks. The attention mechanism computes representations by considering global dependencies across the entire sequence, which is defined as:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{D_k}}\right) V \tag{2.1}$$

where:

  - Q, K, V are the query, key, and value matrices.
  - Dk is the dimensionality of the keys.

The encoder generates a sequence of hidden representations $\mathbf{z}$:

$$\mathbf{z} = \text{Encoder}(\mathbf{x}) \tag{2.2}$$

Position Dependent Summarizer

The PDS bridges the gap between the continuous acoustic input and the discrete token sequence by summarizing the encoder outputs to match the token sequence length.

- Let L be the target token sequence length.
- The PDS module generates token-level representations:

$$\mathbf{h}_i = \text{PDS}(\mathbf{z}, \text{PE}(i)), \quad i = 1, 2, \ldots, L \tag{2.3}$$

The positional encoding PE(i) is defined as:

$$\text{PE}(i, 2j) = \sin\left(\frac{i}{10000^{2j/d_m}}\right), \quad \text{PE}(i, 2j+1) = \cos\left(\frac{i}{10000^{2j/d_m}}\right) \tag{2.4}$$

where d_m is the model dimension, and j is the index.

Decoder

The decoder predicts the output token sequence by utilizing the token-level representations from the PDS.

- For each token position i, the decoder computes the probability distribution over the vocabulary:

$$P(y_i|\mathbf{x}) = \text{Softmax}(\text{Linear}(\mathbf{h}_i)) \tag{2.5}$$

The model is trained to minimize the cross-entropy loss over the entire sequence:

$$\mathcal{L}(\theta) = -\frac{1}{NL} \sum_{(x,y)\in D} \sum_{i=1}^{L} \log P(y_i|\mathbf{x};\theta) \tag{2.6}$$

where:

- D is the training dataset with N examples.
- L is the length of the output sequence.
- θ represents the model parameters.

Experimental Evaluation

Experiments were conducted on the AISHELL-1 dataset, a publicly available Chinese speech corpus. The LASO model demonstrated impressive performance, achieving a Character Error Rate (CER) of 6.4% on the test set. This result outperformed several autoregressive baselines, including the Transformer model, which achieved a CER of 6.7%.

Notably, LASO showed a significant improvement in inference speed. The average latency was reduced to just 21 milliseconds per utterance, making it approximately 50 times faster than traditional autoregressive models. This substantial speed-up was achieved due to LASO's non-autoregressive architecture, which allows for parallel token generation without the need for beam search.

These results highlight LASO's potential in low-latency applications, such as voice assistants and live transcription, where fast and efficient speech recognition is crucial.

2.2 Future Cost Modeling in Neural Machine Translation(Duan et al.)[2]

2.2.1 Introduction

Traditionally, translations produced by neural machine translation (NMT) rely solely on historical context. To improve translation accuracy and coherence, we can predict the cost of creating future target words by implementing a "future cost" technique.

2.2.2 Key Contributions

- **Future Cost Mechanism**: Adds an additional loss to training by predicting the cost of the subsequent target word.
- **Enhanced Models**: Two models utilize this mechanism:
  - **Model I**: Incorporates future cost during training.
  - **Model II**: Extends Model I by integrating future cost into the decoding process, refining the translation output.
- **Experimental Results**: BLEU scores on the English-German, English-French, and Chinese-English datasets showed notable gains.

2.2.3 Methodology

Future Cost Representation

In traditional PBSMT, the "future cost" represents an estimate of the difficulty of translating the remainder of a source sentence. But here, the future cost for every target word is dynamically assessed depending on the current translation context in order to apply this idea to NMT.

A GRU-inspired structure is used to calculate this representation, focusing on the impact of the current word on the next word to be generated. This structure utilizes both the context vector from the previous target words and the source sentence to predict the "future cost representation" for the next word:

$$R_i = \sigma(W_r \cdot E[y_i] + U_r \cdot H_i^N) \tag{2.7}$$

$$Z_i = \sigma(W_z \cdot E[y_i] + U_z \cdot H_i^N) \tag{2.8}$$

$$S_i = \text{ReLU}(W \cdot E[y_i] + U \cdot (R_i \odot H_i^N)) \tag{2.9}$$

$$F_i = Z_i \odot S_i + (1 - Z_i) \odot H_i^N \tag{2.10}$$

where E is the target embedding matrix, W and U are trainable matrices, and σ is the sigmoid function. Here, F_i represents the future cost for word y_i. where E is the target embedding matrix, W and U are trainable matrices, and σ is the sigmoid function. Here, F_i represents the future cost for word y_i.

Model I: Future Cost in Training

Model I utilizes the future cost representation as an additional loss term during training to encourage the model to consider the potential impact of each target word on subsequent words.

Training Objective: The primary loss function is defined as the sum of the standard cross-entropy loss and a future cost loss:

$$J(\theta) = L(\theta) + \lambda \cdot F(\theta), \tag{2.11}$$

where $L(\theta)$ is the cross-entropy loss, $F(\theta)$ represents the future cost, and λ is a hyperparameter controlling the weight of the future cost.

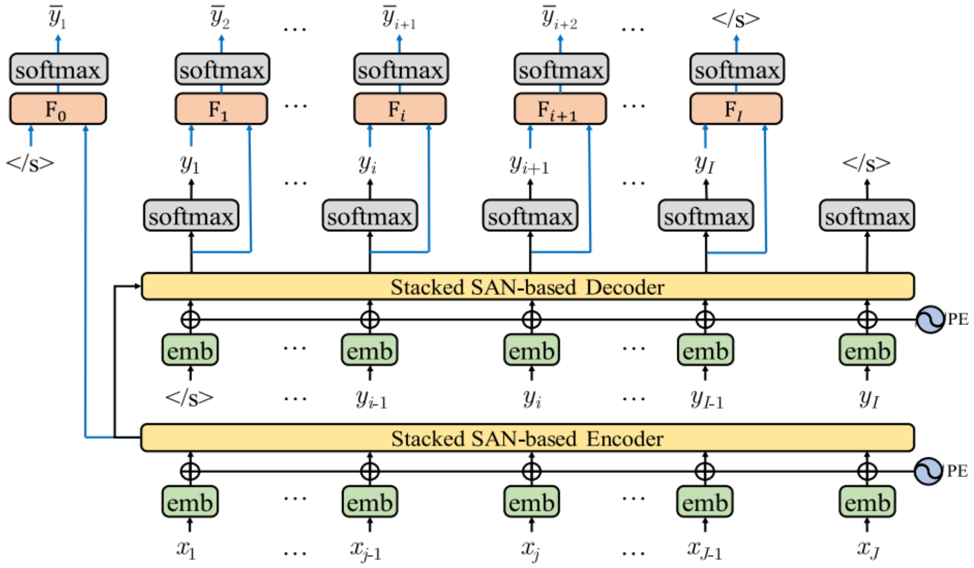


Figure 2.2: Architecture of Model I, showing the additional loss term for future cost during training.

Model II: Future Cost in Decoding

Model II builds on Model I by incorporating the future cost directly into the decoding process, enhancing the generation of target words by considering the estimated cost of the next word at each step.

Integration in Decoding: At each time-step i, the future cost representation from the previous step F_{i-1} is combined with the context representation H_i^N using a gate mechanism to form a fused representation:

$$g_i = \sigma([H_i^N : F_{i-1}]W_g) \tag{2.12}$$

$$H_i^{N'} = H_i^N + g_i \odot F_{i-1} \tag{2.13}$$

where W_g is a trainable parameter and $\odot$ represents element-wise multiplication. This fused representation $H_i^{N'}$ is then used to predict the next word.

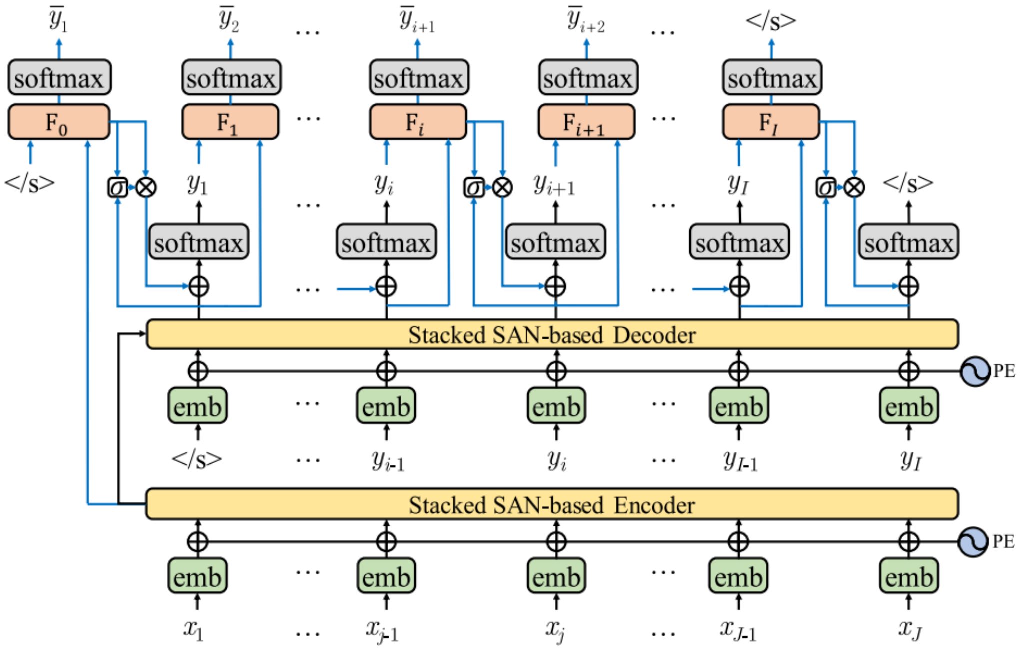


Figure 2.3: Architecture of Model II, illustrating the integration of future cost into decoding.

Summary of Models

- **Model I**: Focuses on using future cost as an additional loss during training to improve translation quality.
- **Model II**: Extends Model I by including future cost in both training and decoding, leading to better predictions at each time-step.

2.2.4 Results

Experimental results demonstrate that the proposed models outperform traditional NMT models, especially in terms of BLEU scores and stability on longer sentences.

Model	EN-DE BLEU	EN-FR BLEU	ZH-EN BLEU
Trans.base	27.03	39.51	32.16
+ Model I	27.56	40.12	32.74
+ Model II	27.64	40.31	33.02

Table 2.1: Performance comparison of baseline and proposed models on BLEU scores

2.2.5 Conclusion

This work demonstrates the effectiveness of a future cost mechanism in NMT, particularly in improving BLEU scores and stability across multiple translation tasks. Model II's integration of future cost into decoding results in the best performance, making it a promising approach for enhancing NMT.

2.3 Audio Visual-Automatic Speech Recognition(D. Serdyuk et al.)[3]

The topic describes the integration of visual signals into automatic speech recognition (ASR), creating audio-visual ASR (AV-ASR) systems that improve robustness, particularly in noisy environments. Traditional ASR systems process audio, but AV-ASR leverages visual cues, like mouth movements, which help in interpreting speech, unaffected by ambient noise. Lip reading, a subset of AV-ASR, relies solely on these visual cues.

Historically, AV-ASR visual feature extraction used 3D convolutional networks, such as VGG models, designed to handle both spatial and temporal data. Recently, however, transformer-based models have emerged as powerful alternatives. [3]

2.3.1 AV-ASR System Architecture

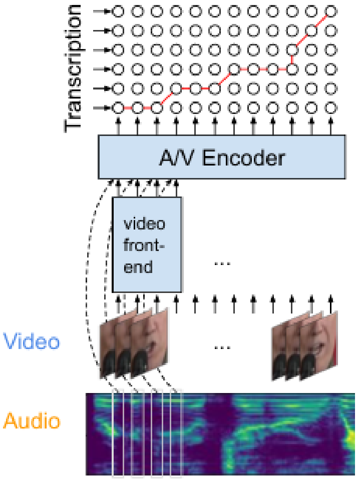


Figure 2.4: Audio-visual encoder combines audio and visual cues for transcription.

This document provides an overview of the architecture for the end-to-end audio-visual automatic speech recognition (AV-ASR) and lip reading model.

2.3.2 Model Components

- **Acoustic Features**: The audio signal is processed into mel filterbank features. The audio, sampled at 16 kHz, is divided into 25 ms frames with a 10 ms step. This produces 80-channel mel filter energies per frame. Every 3 consecutive frames are folded together, yielding a 240-dimensional feature vector every 30 ms, synchronized to a frame rate of approximately 33.3 Hz.

- **Visual Features**: Video data, with frame rates ranging from 23 to 30 fps, is synchronized to the audio frame rate through resampling. A 128 × 128 region around the speaker's mouth is cropped from each frame and normalized to range [-1, 1]. A video front-end (e.g., 3D ConvNet) extracts visual features from these mouth regions, resulting in a visual tensor representation.

- **Modality Fusion**: The visual and audio features are concatenated along the temporal dimension, creating a unified tensor of audio-visual features:

  where A is the audio feature tensor, V is the visual feature tensor, M is the batch size, T is the number of time steps, and D_A and D_V are the dimensions of the audio and visual features, respectively.

- **Encoder**: The combined audio-visual features F are fed into a 14-layer transformer encoder, which applies self-attention mechanisms to process sequential information across time.

- **Decoder**: The encoded features are passed to an RNN Transducer (RNN-T) decoder, which consists of two LSTM layers with 2048 units each to produce character tokens as transcription output.

This architecture leverages both visual and acoustic features for enhanced AV-ASR and lip reading performance.

2.3.3 Methods

(2+1)D Convnet Baseline

In the baseline model, a convolutional network (ConvNet) is used as the video front-end to process the visual data in AV-ASR experiments. This baseline is based on a prior design, which uses a 3D convolutional neural network (3D ConvNet) to capture both spatial and temporal aspects of video data. A 3D ConvNet applies a 3D kernel across both spatial (height and width of the image) and temporal (time) dimensions simultaneously. However, pure 3D convolutions are computationally expensive.

Key Modification: 3D Convolution Decomposition To improve efficiency, the baseline model modifies the 3D convolutions by decomposing each 3D kernel into two separate kernels—one for spatial dimensions and one for the temporal dimension. This is often referred to as a "(2+1)D convolution."

Specifically: - A typical 3D kernel of size $[3, 3, 3]$ (where all dimensions are processed at once) is replaced by two kernels: - A spatial kernel $[1, 3, 3]$ that only processes spatial features (width and height) of a frame. - A temporal kernel $[3, 1, 1]$ that then processes information across frames over time.

This decomposition effectively splits the original 3D convolution into two steps, one for spatial features and one for temporal features, which reduces computational cost while still capturing both spatial and temporal information.

Using this (2+1)D decomposition, the baseline ConvNet is structured as follows: - A VGG-like architecture is used with 5 main layers, but after decomposition, this results in a 10-layer network. - Max pooling (a technique that reduces the spatial dimensions of the data) with a size of 2 is applied after every two layers, except for the fourth layer. Max pooling helps further reduce computation by down-sampling the spatial dimensions, which also focuses on the most significant features.

This (2+1)D ConvNet thus provides a more efficient, yet effective, way to extract spatial-temporal features from video frames, serving as a strong baseline for visual processing in AV-ASR tasks.

2.3.4 Video Transformer Front End

The proposed architecture for feature extraction in the AV-ASR model leverages a transformer-based approach inspired by previous works. Here's a breakdown of its components and process:

1. Tubelet Extraction: - The video data is divided into small 3D segments called "tubelets," which are analogous to patches in image processing but with an additional temporal dimension. Each tubelet is a 4-dimensional structure, capturing information

along width, height, time, and color channels. - In this setup, each tubelet has dimensions of $32 \times 32 \times 8$ (width $H = 32$, height $W = 32$, and temporal depth $T = 8$). This temporal depth indicates that each tubelet captures information across 8 consecutive frames. - Tubelets are non-overlapping in the spatial dimensions, meaning they cover unique parts of each video frame without overlap, and are extracted at every time step (with a temporal stride of 1).

2. Flattening and Embedding: - Once the tubelets are extracted, they are flattened into 1-dimensional vectors to prepare them for input into a transformer. - These flattened tubelets are then transformed through an affine projection, which linearly maps each tubelet vector into a higher-dimensional space to create embeddings. This step enables the model to work with high-dimensional representations of visual data.

3. Positional Embeddings: - Because transformers lack inherent sequence or positional awareness, a positional embedding is added to each tubelet embedding. This positional information helps the model understand the spatial and temporal order of the tubelets within the video.

4. Transformer Architecture: - The architecture includes a standard 6-layer transformer, which is commonly used in sequence-based tasks. The transformer uses an 8-headed self-attention mechanism and generates embeddings with 512 dimensions. - The transformer processes each tubelet embedding in parallel, capturing relationships and context across the entire video segment at each time step.

5. Output: - The final step in this architecture takes the first output from the last transformer layer and sends it to the rest of the AV-ASR (audio-visual automatic speech recognition) or lip reading network. This output contains a condensed representation of the visual features, effectively summarizing the relevant information from each tubelet for downstream processing.

From the Diagram

1. **Video Input**: The video is split into small blocks (patches) of frames.

2. **Linear Projection**: These patches are turned into a set of numbers (embeddings) that the transformer can understand.

3. **Transformer Encoder**: The transformer analyzes these patches to learn patterns across time and space.

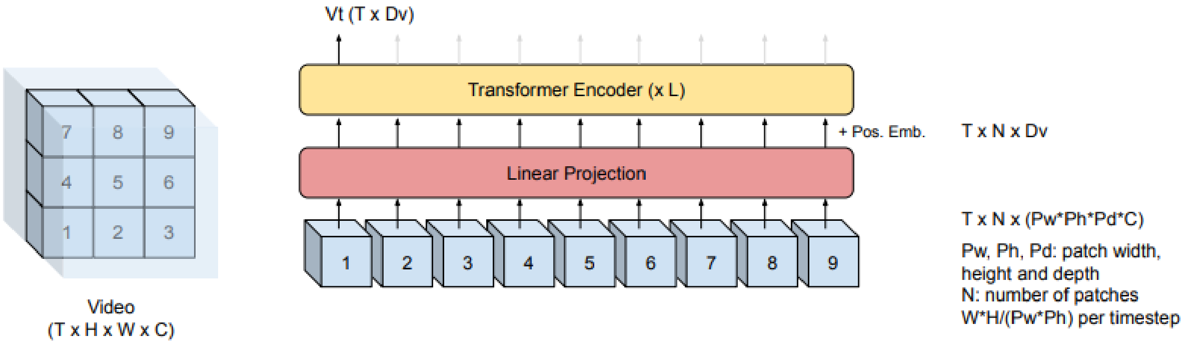

Vt (T x Dv)
Transformer Encoder (x L)
+ Pos. Emb.
T x N x Dv
Linear Projection
T x N x (Pw*Ph*Pd*C)
Pw, Ph, Pd: patch width, height and depth
N: number of patches W*H/(Pw*Ph) per timestep
1
2
3
4
5
6
7
8
9
Video
(T x H x W x C)

Figure 2.5: Transformer model processes video patches to capture spatial-temporal information.

4. **Output**: The result is a set of features that help with tasks like lip reading or speech recognition.

2.3.5 Audio-Visual Automatic Speech Recognition

1. Evaluation on YTDEV18

- **Baseline Comparison:** The performance of the visual transformer front-end (ViT3D) was compared to a VGG-based baseline. The ViT model performed similarly to the baseline when the audio was clear, indicating that audio alone is a strong signal for the model.

- **Noise Testing:** Additive noise (20, 10, and 0 dB) and overlapping noise (random utterances) were introduced to test robustness. The ViT model outperformed the VGG baseline in high-noise conditions (0 dB), but its performance degraded with overlapping noise. This was attributed to a domain shift, as training data was augmented with only additive noise.

- **Importance of Visual Information:** An audio-only model performed worse in noisy conditions, showing that visual input helps with recognition in challenging audio environments.

2. **Evaluation on LRS3-TED**

- LRS3-TED, a dataset of TED talk recordings, presents simpler conditions (high audio/video quality and centered speaker) compared to YTDEV18. The model performed better here, with the ViT model surpassing the convolutional baseline.

- **Fine-Tuning:** Fine-tuning was performed by mixing the model's training data with the LRS3-TED training set and adjusting the learning rate. This improved the model's performance, making it comparable to state-of-the-art results from other studies.

2.3.6 Conclusion

Better Performance: The model does better than the older convolution-based model for lip reading. Dataset Limitations: The model couldn't be tested on some popular datasets like LRS2 and LRW because of licensing issues. Future Plans: The next steps include improving the model with a different front-end (conformer) and expanding it to handle live (online) speech recognition and multiple speakers.

2.4 Arabic Speech Synthesis using Deep Neural Networks(A. Ali et al.)[4]

2.4.1 Introduction

Arabic speech synthesis presents distinct challenges due to the language's complex phonetic structure, diverse dialects, and context-dependent pronunciation. Arabic, with its rich linguistic heritage, has multiple dialects and a unique set of phonemes that vary widely in pronunciation depending on accents, regional dialects, and situational contexts. Traditional text-to-speech (TTS) models often struggle to accurately capture these nuances, resulting in synthesized speech that may sound unnatural or lack authenticity. In particular, Arabic's script does not always display short vowels, which are essential for precise pronunciation and intonation. This makes it difficult for conventional TTS models, to fully capture the language's phonetic richness.

To address these challenges, this research leverages phoneme-based synthesis, breaking down speech into its fundamental units, or phonemes. By synthesizing Arabic at the phoneme level, TTS models can provide clearer, more natural-sounding speech that mirrors human nuances. This focus on phonemes ensures that the synthesized voice aligns more closely with the authentic Arabic spoken by native speakers, making it suitable for applications such as language learning tools, accessibility software, tailored for Arabic speakers.

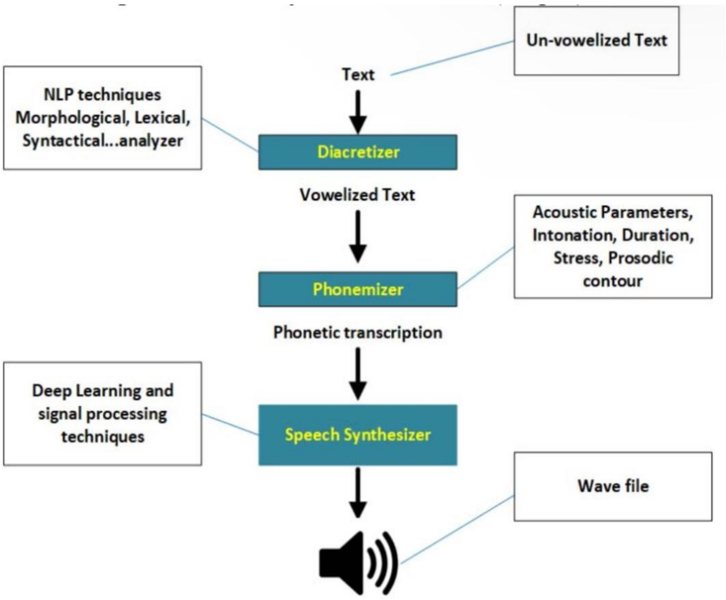

Un-vowelized Text
Text
NLP techniques Morphological, Lexical, Syntactical...analyzer
Diacretizer
Vowelized Text
Acoustic Parameters, Intonation, Duration, Stress, Prosodic contour
Phonemizer
Phonetic transcription
Deep Learning and signal processing techniques
Speech Synthesizer
Wave file

Figure 2.6: Process flowchart

2.4.2 Deep Neural Networks in Arabic TTS: Enhancing Naturalness and Quality

The core algorithm employed in this research relies on Deep Neural Networks (DNNs), which offer powerful capabilities in modeling the complex dependencies and variations in Arabic phonetics. DNNs enable the model to capture subtle language features and dependencies, allowing for a deeper understanding of context and intonation that traditional rule-based or concatenative TTS methods struggle to achieve. DNN-based models are particularly beneficial for Arabic due to their flexibility in learning and generalizing phonetic variations without relying solely on fixed rules. By training on large Arabic datasets, the DNN model learns to associate phonetic inputs with their corresponding acoustic outputs accurately, improving both intelligibility and naturalness.

The effectiveness of the DNN-based Arabic TTS model is assessed through subjective evaluation metrics, primarily the Mean Opinion Score (MOS), a widely used metric for gauging the quality of synthesized speech. MOS evaluation involves human listeners rating the quality of the audio output, providing insights into how accurately the model replicates natural Arabic speech and whether it meets human expectations for pronunciation, rhythm, and intonation. This evaluation ensures that the DNN-based TTS system not only achieves technical precision in its acoustic representation but also delivers a high-quality listening experience that resonates with native Arabic speakers. Through this approach, the study highlights how advanced neural architectures can be tailored to meet the specific demands of Arabic speech synthesis.

2.4.3 Methodology

Figure 2.6 describes an overview of the process,where The TTS synthesis process begins with un-voweled text input, which lacks vowel diacritics or markers, making the text incomplete in terms of pronunciation indicators. This un-voweled text is then passed to the Diacritizer, where Natural Language Processing (NLP) techniques, such as morphological, lexical, and syntactical analysis, are applied to add the appropriate diacritical marks. This step results in voweled text, where the necessary vowel markers provide a guide for correct pronunciation. The voweled text moves on to the , which converts it into a phonetic transcription—a representation of the pronunciation in phonemes, the smallest units of sound. The phonemizer also includes acoustic parameters, such as intonation, duration, stress, and prosodic contours, ensuring that the generated speech will sound natural and expressive. Next, the Speech Synthesizer uses deep learning and signal processing techniques to transform the phonetic transcription into an audible waveform, creating the actual spoken audio. This step produces a wave file output, the final product, which contains the synthesized speech as an audio file that can be played back to deliver the spoken version of the original text.

1. **Data Collection and Preparation**

   The foundation of any deep learning model is the data used for training. In this case, the model used a substantial amount of high-quality Arabic speech data. This typically involves

   - **Arabic Speech Recordings:** A large dataset of Arabic speech recordings covering a wide range of speech patterns, including different phonetic structures, accents, and tones to ensure the model captures the full breadth of the language.

- **Text Annotations:** Each speech recording is paired with corresponding text annotations which is crucial for training the model, as the DNN learns to map the textual input to corresponding speech features.
- **Diverse Data:** The dataset includes different dialects of Arabic focusing mainly on Modern Standard Arabic (MSA).

2. Feature Extraction

Once the dataset is prepared, the next step is to extract features from the raw speech data. The neural network does not process the raw audio directly but instead works on specific features extracted from the speech signal. Common features include:

- **Pitch:** The highness or lowness of the voice, affected by intonation and emotion, making speech sound natural.
- **Duration:** The length of time each sound is held, ensuring speech flows naturally.
- **Spectrum:** The voice quality, capturing the tone and character of speech based on energy at different frequencies.

These features are fed into the neural network to teach it how to convert text into natural-sounding speech.

3. Neural Network Architecture

- **Input Layer:** The text is converted into phonemes or characters and then transformed into numerical vectors through embedding, making it understandable for the neural network.
- **Hidden Layers:** The network contains layers that process the input. RNNs(Recurrent Neural Networks) or LSTMs(Long Short-Term Memory networks) are used here, which are good for handling sequential data like speech.

  - **RNNs:** These process speech one element at a time, keeping track of what came before to understand the sequence. As depcited in the figure, RNN
  - **LSTMs:** A more advanced RNN that can retain information over longer sequences, useful for capturing context in speech.

- **Output Layer:** The network generates predictions for key speech features (pitch, duration, spectrum), which are used to create the final synthesized speech sound.

4. **Training the Model**

- **Loss Function:** The model computes the difference between the predicted features and the actual features extracted from the speech recordings. The most common loss functions used in speech synthesis include mean squared error (MSE), which measures how close the predicted features are to the real ones.
- **Back propagation:** The model uses back propagation to adjust its internal parameters (weights) based on the loss function. Over time, the model learns the mapping from text to speech features with increasing accuracy.
- **Optimization:** Optimization algorithms such as Adam or SGD (Stochastic Gradient Descent) are used to minimize the loss and improve the model's performance.

5. **Speech Synthesis (Inference)**

Once the model is trained, it is ready to synthesize speech. During inference (the stage when the model is actually used to generate speech), the following steps occur:

- Text Input: A sequence of Arabic text is fed into the model.
- Feature Prediction: The trained DNN processes the text and predicts the corresponding speech features (pitch, duration, spectrum).

- Waveform Generation: The predicted features are then passed to a vocoder (a tool that converts speech parameters into audible sound) to generate the final speech waveform. One popular vocoder used in deep learning-based TTS systems is WaveNet, which generates high-quality, natural-sounding speech based on the predicted features.

Deep Learning-Based Speech Synthesis

Figure 2.7 represents a basic deep neural network (DNN) architecture, commonly used in machine learning applications. Following is a breakdown of its components:

Input Layer: The leftmost layer represents the input layer, where data enters the network. Each circle (node) in this layer corresponds to an input feature, labelled as X1, X2, and X3.

Hidden Layers: The middle layers are called hidden layers. They consist of multiple neurons (nodes) organized across multiple layers. In this case, there are several hidden layers, each with a varying number of nodes. These hidden layers process and transform the input data using weights and activation functions, allowing the network to learn complex patterns and representations.

Output Layer: The rightmost node represents the output layer, labelled as Y. This is where the network produces the final prediction or output based on the learned patterns from the hidden layers.

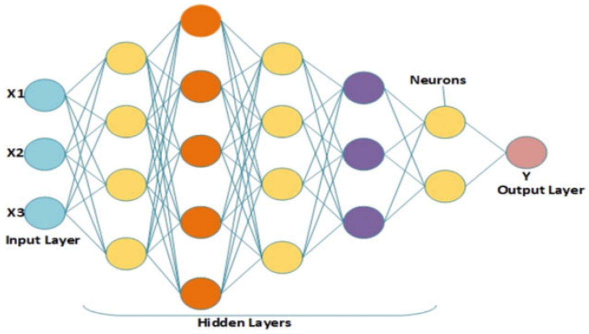


Figure 2.7: Deep Learning Network

Deep learning approaches, such as Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks, offer improvements in speech synthesis by learning from larger amounts of data and improving with training size.

- **RNN + LSTM:**
  - RNNs are designed to handle sequential data, while LSTMs are a type of RNN that retain long-term dependencies, making them well-suited for tasks like speech synthesis where context and timing matter. As depicted in the figure, RNN connections are between nodes and form a directed graph along a sequence and thus can use their internal state (memory) to process sequence of inputs.

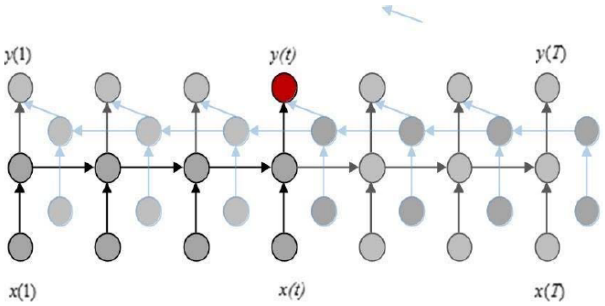


Figure 2.8: Recurrent Neural Networks

  - LSTM units contain input, output, and forget gates, allowing the network ”remember” important aspects of the speech sequence. LSTM blocks are building units for RNN layers. As seen in the figure, it is made up of a cell, an input, an output, and forget gates. Because the cell is in charge of ”remembering” values over arbitrary time periods, the word ”memory” in LSTM refers.

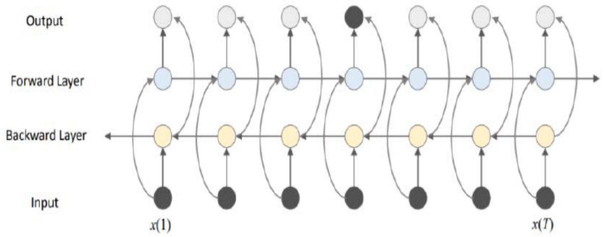


Figure 2.9: Long Short-Term Memory networks

- **Transition to Deep Learning**

  Figure 2.10 illustrates the architecture of Tacotron 2 with WaveNet, an advanced Text-to-Speech (TTS) model that synthesizes highly natural speech from text. The process begins with input text, which is converted into numerical vectors using character embedding. These embeddings are processed

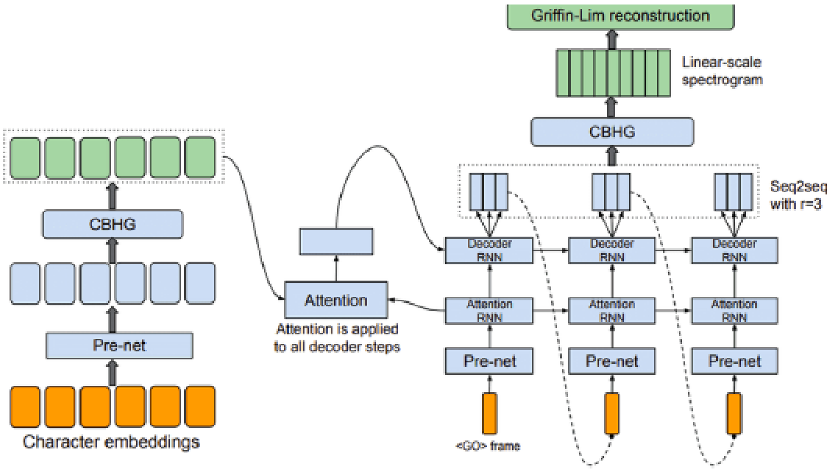


Figure 2.10: Tacotron model architecture

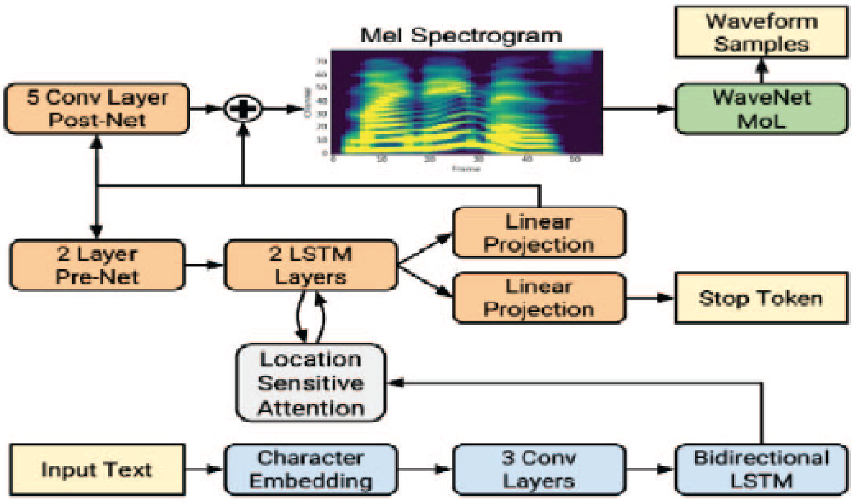


Figure 2.11: Tacotron 2 model architecture

through a series of convolutional layers and a bidirectional LSTM network in the Text Pre-Net, extracting important features and capturing context from both sides of each character. The model then uses a Location Sensitive Attention mechanism to align text features with corresponding parts of the speech, ensuring that the pronunciation and timing are accurate. The aligned features are passed through two LSTM layers to account for long-term dependencies in speech, and a linear projection layer transforms them into an initial Mel spectrogram, a visual representation of sound frequencies over time. To improve quality, a 5-layer Convolutional Post-Net further refines the spectrogram.

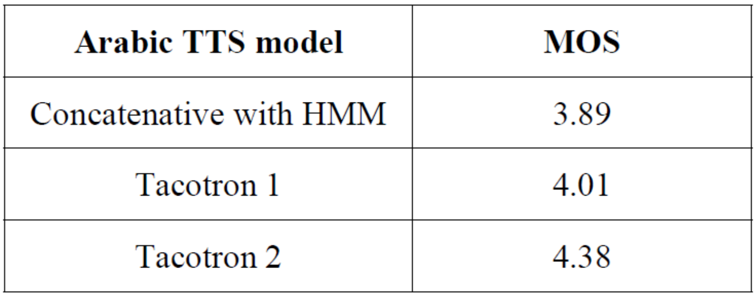
This refined Mel spectrogram is then fed into WaveNet, a neural vocoder that synthesizes the final audio waveform. WaveNet generates high-quality, natural-sounding speech by modeling the complex patterns in audio, including variations in pitch, tone, and articulation. It uses dilated convolutional layers to capture detailed audio characteristics, producing fluid and lifelike speech. Additionally, the Tacotron 2 model includes a stop token to signal when the speech synthesis should end, ensuring efficiency. Overall, this combination of Tacotron 2 and WaveNet enables the generation of expressive and realistic speech by leveraging deep learning and advanced attention mechanisms.

Tacotron 2 is a deep learning-based model designed for text-to-speech synthesis, where it takes characters as input and generates a mel spectrogram as an intermediary step. Originally, the Tacotron model utilized the Griffin-Lim algorithm to convert the mel spectrogram into a waveform, thereby producing synthetic speech. However, Tacotron 2 introduced a significant improvement by replacing the Griffin-Lim algorithm with a WaveNet vocoder. This vocoder uses neural networks to model waveforms directly, resulting in more natural, high-quality speech synthesis. With WaveNet, Tacotron 2 achieves a more human-like prosody, offering smoother intonation and rhythm, which significantly enhances the subjective audio quality. This improvement is especially noticeable when handling complex Arabic speech patterns, where Tacotron 2 excels in delivering natural-sounding results.

6. Evaluation

To assess the quality of the synthesized speech, the model is evaluated using standard speech synthesis metrics:

- **MOS (Mean Opinion Score):** On a scale of 1 to 5, with 5 representing the most natural-sounding, human listeners evaluate the quality of synthetic speech in this subjective assessment.



Arabic TTS model	MOS
Concatenative with HMM	3.89
Tacotron 1	4.01
Tacotron 2	4.38

Figure 2.12: Mean Opinion Score of 50 participants

- **Objective Metrics:** The quality and intelligibility of the synthesized speech can be evaluated objectively using metrics such as Mel Cepstral Distortion (MCD), Perceptual Evaluation of Speech Quality (PESQ), or Word Error Rate (WER) in addition to subjective evaluation. Mel Cepstral Distortion (MCD) can be calculated using this formula:

$$\text{MCD} = \frac{10}{\ln 10}\sqrt{2\sum_{d=1}^{D}\left(c_d^{(ref)} - c_d^{(syn)}\right)^2}$$

  where:

  - D: The number of mel-cepstral coefficients (typically 13).
  - $c_d^{(ref)}$: The d-th mel-cepstral coefficient from the reference (ground truth) speech.
  - $c_d^{(syn)}$: The d-th mel-cepstral coefficient from the synthesized speech.
  - MCD: The Mel Cepstral Distortion, which measures the spectral difference between the reference and synthesized speech.

2.5 Summary and Gaps Identified

This section provides a detailed summary of the literature reviewed and identifies the gaps that motivate further research.

Title	**Advantages**	**Disadvantages**
LASO: Non-Autoregressive Architecture for Low-Latency Speech Recognition	Enables parallel computation, significantly reduces latency, achieves high accuracy.	Limited applicability to complex sentences due to token independence.
Future Cost Modeling in Neural Machine Translation	Improves translation coherence, enhances performance on multiple language pairs.	Increased computational complexity during decoding; limited experimentation on low-resource languages.
Audio-Visual Automatic Speech Recognition	Enhances robustness in noisy environments; leverages visual cues effectively for lip-reading tasks.	Reduced performance in overlapping noise conditions; requires high-quality synchronized data.
Arabic Speech Synthesis Using Deep Neural Networks	Captures phonetic nuances effectively; improves naturalness and quality in speech synthesis.	Limited application to regional languages with diverse dialects like Malayalam.

Table 2.2: Summary of Literature Reviewed

Despite advancements in speech recognition, machine translation, and text-to-speech synthesis, significant gaps remain in the current state of the art. The scarcity of Malayalam-specific datasets limits model training and evaluation, while existing systems inadequately address dialectal variations and cultural nuances essential for contextual accuracy. Synchronization between translated audio and video poses a major challenge, particularly in maintaining natural flow and timing. Additionally, the lack of scalable and cost-effective solutions restricts the applicability of automated translation systems for regional languages.

2.6 Conclusion

The literature survey explored advancements in automatic speech recognition, machine translation, and text-to-speech synthesis, highlighting their relevance to the development of the Automated Video Translator. Key insights include the potential of non-autoregressive models for real-time performance, the integration of future cost modeling to improve translation coherence, and the advantages of combining audio-visual inputs for enhanced speech recognition in noisy environments.

Chapter 3

Requirements

3.1 Hardware and Software Requirements

- Processor: Intel Core i5
- RAM: 8GB
- GPU: 4GB NVIDIA GeForce GTX 1650
- IDE: Visual Studio Code
- Operating system: Windows 10(or later)
- Tech Stack: Python (3.8)
- Machine Translation Frameworks: TensorFlow

  TensorFlow is an open-source library for machine learning, which gives developers a range of tools to train, and deploy machine learning models effortlessly.
- Libraries: FFmpeg, Spleeter, Librosa, pydub, pyttxs3, SpeechRecognition, MoviePy

  These libraries facilitate audio and video processing within Python
- Frameworks: Flask

  Its a lightweight Python backend web framework that makes it simple for developers to create server-side logic and easily manage databases, and user authentication.

Chapter 4

System Architecture

The Automated Video Translator utilizes a combination of well-defined steps to achieve seamless translation of Malayalam audio into English translated audio and its synchronisation with the actual video.

4.1 Architecture Design

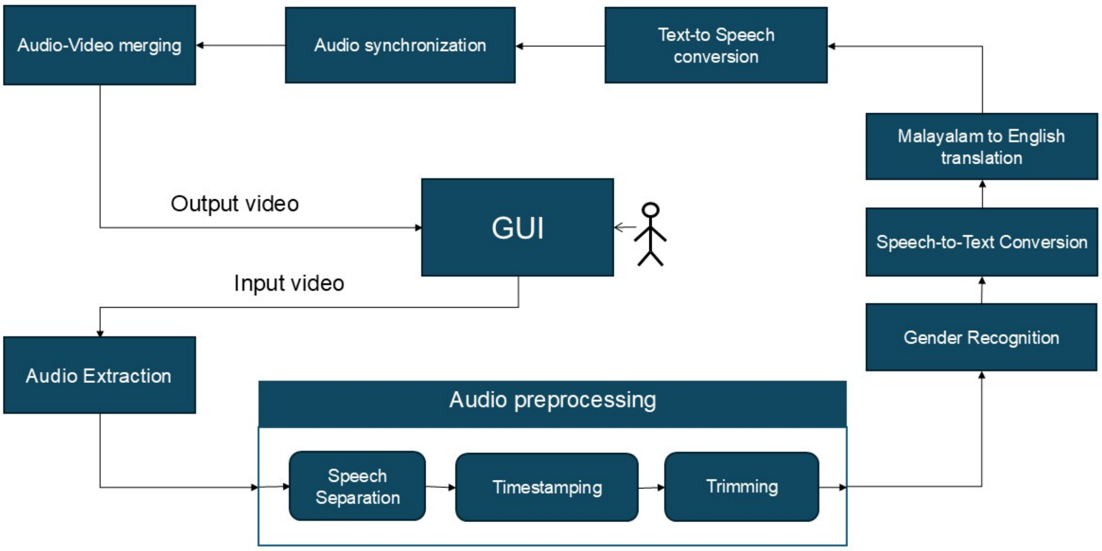


Figure 4.1: Architectural Diagram

The audio is first extracted from the video which is then separated from the accompanying sounds like background music and noise. This audio is then seperated into many audio segments based on the timestamps identified wrt the silence in the audio. These audio segments are converted to English texts which are then converted to corresponding English audios. Finally, the system synchronizes the generated audios with the accompanying sounds using the timestamps detected previously. Lastly the synchronized audio is merged back with the video ensuring accurate timing, natural flow, and retention of background sounds for a seamless translation experience.

4.2 Component Design

4.2.1 Audio Extraction Module

Purpose

Extract the audio from the video files.

Details

This module uses **FFmpeg** to extract the audio track from the input video file. It makes sure that the extracted audio is of high quality to enable accurate and efficient downstream processing.

4.2.2 Audio Preprocessing Module

Purpose

Isolate speech from background sounds, save timestamps of speech segments, and trim the audio accordingly.

Details

Spleeter is used to isolate the speech from background noise and music. **Librosa** assists in finding silent sections to identify the speech segment timestamps. Based on these timestamps, audio is trimmed using **Pydub** to create smaller, targeted clips with only speech.

4.2.3 Gender Recognition Module

Purpose

To identify male and female voice within the audio.

Details

In order to identify the male and female voices in the audio segments, we implement a custom ResNet-based CNN model to train a dataset and provide accurate results. Using this module the speaker authenticity is maintained even in the English audio.

4.2.4 Speech-to-Text Conversion Module

Purpose

Transform the trimmed Malayalam speech into text.

Details

This process is executed with the help of the **SpeechRecognition** library in Python. It translates the preprocessed audio recordings and converts the spoken Malayalam material into written words.

4.2.5 Translation Module

Purpose

Translate the Malayalam text into English.

Details

The translated output is created with the **Google Translate API**. The emphasis is on maintaining contextual meaning while translating to ensure that the resulting English text is an accurate representation of the original Malayalam material.

4.2.6 Text-to-Speech Module

Purpose

Translate the translated English text into speech.

Details

The **pyttsx3** library is used to generate English speech from the translated text. The synthesized audio is made to sound clear and natural, offering a high-quality reading of the translated content.

4.2.7 Audio Synchronization Module

Purpose

Synchronize the newly synthesized English audio with the background sounds.

Details

This module synchronizes the English speech with the original non-speech audio elements, like background effects or music using pydub. The intention is to preserve the original mood and ambiance of the video and swap the speech material.

4.2.8 Audio-Video Merging Module

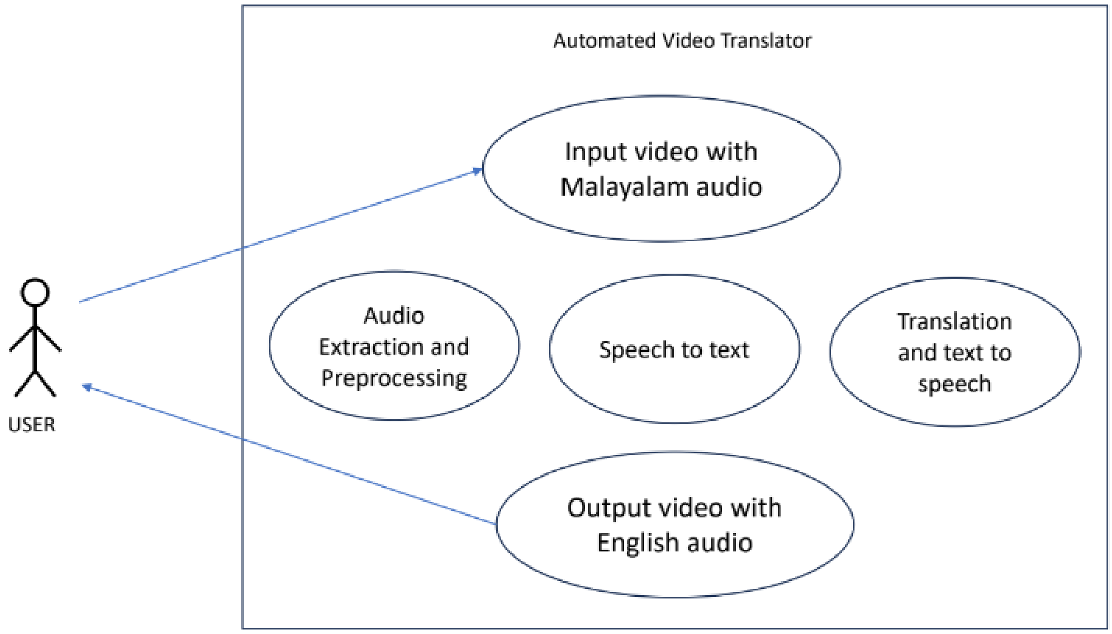
Purpose

Combine the audio back with the original video.

Details

With **MoviePy**, the resultant English audio, synchronized with background elements now is merged with the initial video. This creates the final video with English audio instead of the original Malayalam speech.

4.3 Use Case Diagram


Automated Video Translator
Input video with Malayalam audio
Audio Extraction and Preprocessing
Speech to text
Translation and text to speech
Output video with English audio
USER

This use case diagram represents the Automated Video Translator system, which converts a video with Malayalam audio into a video with English audio.

1. **User:** The primary actor who interacts with the system. The user provides the input video and receives the translated output video.

2. **Input Video with Malayalam Audio:** The user supplies a video that contains audio in Malayalam as the initial input to the system.

3. **Audio Extraction and Preprocessing:** This process extracts the audio from the video and preprocesses it to make it suitable for further steps.

4. **Speech-to-Text:** The system converts the processed Malayalam audio into corresponding text.

5. **Translation and Text-to-Speech:**

    - The recognized Malayalam text is translated into English.
    - The translated English text is then converted back to audio using text-to-speech (TTS) technology.

6. **Output Video with English Audio:** The translated English audio is combined with the original video, creating a video with English audio as the output for the user.

4.4 Sequence Diagram

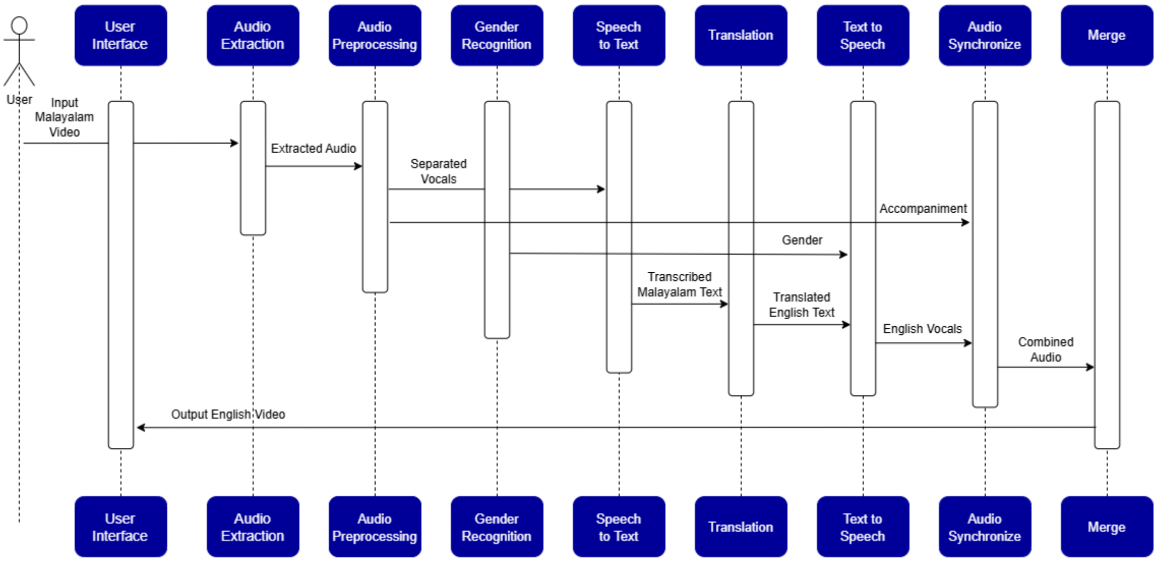


Figure 4.2: Sequence Diagram

4.5 Gantt Chart

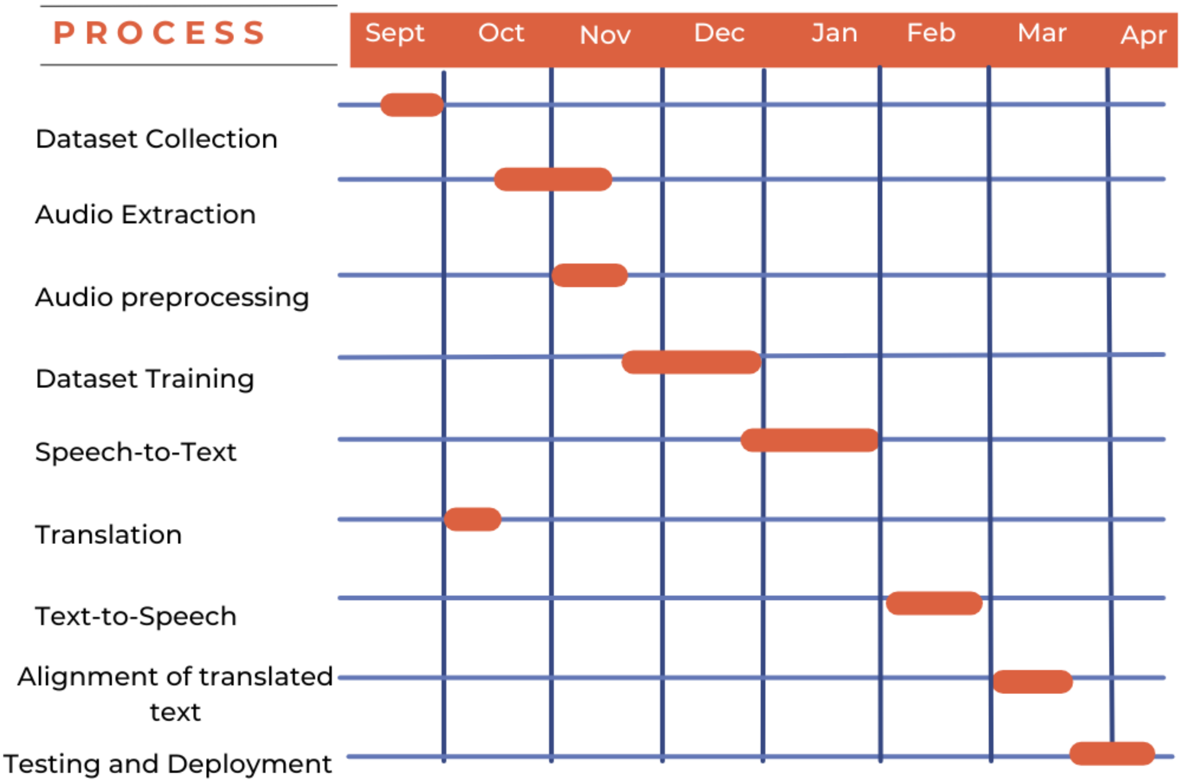


Figure 4.3: Gantt Chart

Chapter 5

System Implementation

This chapter explains in detail the methodology applied in this project and how it was implemented. The whole project was developed in Python. Flask was utilized for the backend server, and the User Interface was rendered using HTML, CSS, and JavaScript. Flask is a lightweight Python framework for assisting in the development of web applications, but HTML gives the layout, CSS adds the styling, and JavaScript provides the interactivity to the user interface.

5.1 Proposed Methodology

5.1.1 Audio Extraction

The Audio Extraction Module was specifically implemented to pull audio out from video files. The video is first loaded into the system through an interactive Frontend made using HTML, CSS and JavaScript. FFmpeg, a Python library, is used to fetch the audio stream from the input video while making sure that the audio extracted was of high quality. This was necessary to facilitate accurate and efficient processing in the next phases of the system.

5.2 Audio Preprocessing

Audio Preprocessing Module was created to separate the speech from noises, detect the timestamps of the speech segments, and trim the audio. Spleeter separates the speech from the accompanying sounds like background noise and music so that there could be clearer focus on the content being spoken. Librosa is used in the detection of silent parts of the audio to properly mark the beginning and end of every speech segment. These timestamps were used to then cut the audio with Pydub, creating smaller focused clips containing merely the speech bits, easier for subsequent processing.

5.2.1 Gender Recognition

For Gender Recognition, a custom ResNet-based CNN for audio classification was developed. A model was trained using a large dataset of male and female audio samples, each containing 2000 WAV files. The model achieved 95% accuracy, significantly improving the precision of gender recognition. The input to the model is a Malayalam audio file, and the output is a string indicating 'male' or 'female'. After identifying the gender using the model, 1 is returned for male and 2 for female. Audio segments shorter than 0.6 seconds are ignored, assuming they do not contain any meaningful speech. For such segments, 0 is returned. The module was significant in maintaining the authenticity of the speaker, as the detected gender was kept in the synthesized English audio, providing a more natural and uniform listening experience.

A special mention to VoxCeleb for providing an excellent open-source dataset. The VoxCeleb dataset is available for commercial and research use under the Creative Commons Attribution 4.0 International License.

The original audio files in the dataset were in M4A format. However, most audio processing libraries (such as Librosa, PyDub, SpeechRecognition, Whisper, etc.) and models are optimized for WAV input. So, the audio files are converted to WAV format for model training. Additionally, we ensured that the male and female datasets were balanced, with 2000 files each.

5.2.2 Speech-to-Text Conversion

The Speech-to-Text Conversion Module was developed to transform the trimmed Malayalam speech into text. This process was carried out using the SpeechRecognition library in Python, a popular open-source Python library used for accurately converting the spoken Malayalam content into written form. This textual representation served as the foundation for the subsequent translation and synthesis stages of the system.

5.2.3 Translation

The Translation Module was intended to translate the identified Malayalam text into English. This was done through the use of the Google Translate API, which offered effective and consistent translation functionality. Particular care was taken to ensure that the contextual meaning of the original material was maintained.

5.2.4 Text-to-Speech

For converting English text to audio, the pyttsx3 library is used. pyttsx3 is a Python text-to-speech conversion library that uses the system's built-in speech engine. It supports voice customization, including male and female voices. The value returned by the gender recognition module, the length of the audio segment, and the translated text are passed to the text-to-speech module. Using the length of the audio segment and the number of words in the translated text, the appropriate output speed (in words per minute) is calculated. The voice (male/female) is selected based on the value returned by the gender recognition module.

5.2.5 Audio Synchronization

The Audio Synchronization Module was created to synchronize the newly synthesized English audio with the background sounds in the original video. The module ensures that English speech was accurately aligned with non-speech audio items like background music or sound effects. This is done using pydub, a library in python.

5.2.6 Audio-Video Merging

The Audio-Video Merging Module was tasked with merging the final English audio with the original video. With MoviePy, the synchronized English speech-coupled with the retained background audio elements—was merged back into the video seamlessly. This left a final output where the original Malayalam audio was successfully replaced with English, without compromising the overall audiovisual integrity and timing of the original content.

5.3 User Interface Design

The user interface of our project was carefully crafted with HTML, CSS, and JavaScript to provide a well-structured layout for a smooth user experience. It enables users to enter a Malayalam video, which is processed by the system to generate an English video as output. The interface is made to be straightforward and easy to use, making it possible for users to simply upload their source video and get a translated version.

5.4 Implementation Strategies

The entire project was built using Python. The following sections contain code snippets for the various modules.

5.4.1 Audio Extraction

```
Start
Set input_video ← ”D:/My Folder/Project/Major-Project/Code/inp3.mp4”
Call FFmpeg to:

    • Take input from input_video

    • Extract audio from the video

    • Save the output as ”D:/My Folder/Project/Major-Project/Code/out.mp3”

    • Overwrite the output file if it already exists

Print ”Audio Extracted.”
Wait for 1 second
End
```

5.4.2 Audio Preprocessing

Splitting the audio:

```
Start
Initialize Separator with 2 stems (vocals and accompaniment)
Set input_audio ← ”D:/My Folder/Project/Major-Project/Code/out.mp3”
Set output_dir ← ”output”
if output_dir does not exist then
    Create output_dir
end if
Call separator to:
    • Separate input_audio into stems
    • Save separated files into output_dir
Print "Separated files are saved in output"
End
```

Detecting timestamps:

```
Function preprocess_audio(file_path)
Load audio from file_path with:
    • Sample rate = SAMPLE_RATE
    • Duration = DURATION
if audio length ¡ SAMPLE_RATE × DURATION then
    Pad audio with zeros at the end to match required length
else
    Trim audio to required length
end if
Extract MFCC features using:
    • SAMPLE_RATE
    • Number of MFCCs = N_MFCC
if number of MFCC time steps ¡ TARGET_WIDTH then
    Pad MFCC matrix with zeros to reach TARGET_WIDTH
else
    Trim MFCC matrix to TARGET_WIDTH
end if
Add batch and channel dimensions to MFCC matrix
Return the processed MFCC matrix
End Function
```

Trimming:

```
Start
Set audio_file ← "D:/My Folder/Project/Major-Project/Code/output/out/vocals.wav"
Set output_dir ← ”output_dialogues”
Set silence_thresh ← -40 {Silence threshold in dB}
Set min_silence_duration ← 1 {Minimum silence duration in seconds}
Set pitch_change_thresh ← 14000 {Pitch change detection threshold}
if output_dir does not exist then
    Create output_dir
end if
Call detect_dialogue_segments_with_pitch with:
    • audio_file
    • silence_thresh
    • min_silence_duration
    • pitch_change_thresh
Store the result in dialogue_segments
Call trim_audio_by_segments with:
    • audio_file
    • output_dir
    • dialogue_segments
Print dialogue_segments
Print ”Saved <number of segments> dialogue segments in ’output_dir’”
End
```

5.4.3 Gender Recognition

```
Function predict_gender(file_path, model, label_encoder)
Call preprocess_audio(file_path) and store result in mfcc
Call model.predict(mfcc) and store result in pred_probs
Get index of highest probability from pred_probs
Map index to label using label_encoder, store in pred_class
Get maximum probability value from pred_probs, store in confidence
```

```
Return pred_class, confidence
End Function

Function detect_gender(audio_file, model, label_encoder)
Call predict_gender(audio_file, model, label_encoder) and store result in gender, confidence
Print ”Single file prediction:”
Print file name from audio_file
Print predicted gender
Print confidence as percentage
if gender is ”Male” then
    Print ”Identified Gender: Male”
    Return 1
else
    Print ”Identified Gender: Female”
    Return 2
end if
End Function
```

5.4.4 Speech-to-Text Conversion

```
Function audio_to_text(audio_file_path)
Initialize speech recognizer
Call split_audio(audio_file_path) and store result in chunks
Set full_text to empty string
for each chunk in chunks with index i do
    Set chunk_path $\leftarrow$ ”chunk” + i + ”.wav”
    Export chunk to chunk_path in WAV format
    With chunk_path as audio source:
        Record audio using recognizer
        Try
            Recognize speech using Google Web Speech API (language = ”ml-IN”)
            Append recognized text and a space to full_text
        Except if speech not understood:
            Print "Chunk i: Google Web Speech API could not understand the audio."
        Except if request fails:
            Print "Chunk i: Could not request results from Google Web Speech API"
end for
Print "Malayalam Text: " + full_text
Return full_text (stripped of leading/trailing whitespace)
End Function
```

5.4.5 Translation

```
Function Mal2Eng(Maltext)
Set target_lang $\leftarrow$ "en" {English language code}
Set url $\leftarrow$ Google Translate API endpoint with API_KEY
Set data to:
    • "q" = Maltext
    • "target" = target_lang
Send POST request to url with data
Receive response from API
Extract translatedText from response JSON
Replace all occurrences of &#39; with ' in translatedText and store in Engtext
Return Engtext
End Function
```

5.4.6 Text-to-Speech

```
Function eng2aud(text, dialogue_len, gender, j)
Initialize text-to-speech engine
Split text into words
Calculate num_words as the number of words
Calculate $speech\_rate = (60 / dialogue\_len) \times num\_words$
Get list of available voices from the engine
```

```
for each voice in voices with index do
    Print voice index, name, and ID
end for
Set engine voice to the voice corresponding to *gender*
Set engine speech rate to *speech_rate*
Set engine volume to 1.0 (maximum)
Set output file path to ”final_dialogues/dialogue_segment_” + (j + 1) + ”.wav”
Save synthesized speech of *text* to the output file
Run the engine and wait for speech synthesis to complete
End Function
```

5.4.7 Audio Synchronization

```
Load accompaniment audio from file "accompaniment.wav"
for each index $i$ from 0 to $length(dialogue\_segments) - 1$ do
    Load vocal audio from file "dialogue_segment_(i+1).wav"
    Calculate $timestamp = dialogue\_segments[i][0] \times 1000$ {Convert to milliseconds}
    Overlay vocal audio onto accompaniment at position = $timestamp$
end for
Export the final combined audio to "combined_output.wav" in WAV format
Print ”Vocals and accompaniment combined”
```

5.4.8 Audio-Video Merging

```
Load video from file path stored in inputvideo
Load new audio from "combined_output.wav"
Set the video’s audio to the new audio
Export the updated video to "D:/My Folder/Project/Major-Project/Code/Frontend/output/
    using codec "libx264"
Print "Output Video Generated"
```

5.5 Chapter Conclusion

In this chapter methodology was explained elaborately. The system combines advanced tools with personalized speech-to-text models to automatically translate Malayalam into English audio.

Chapter 6

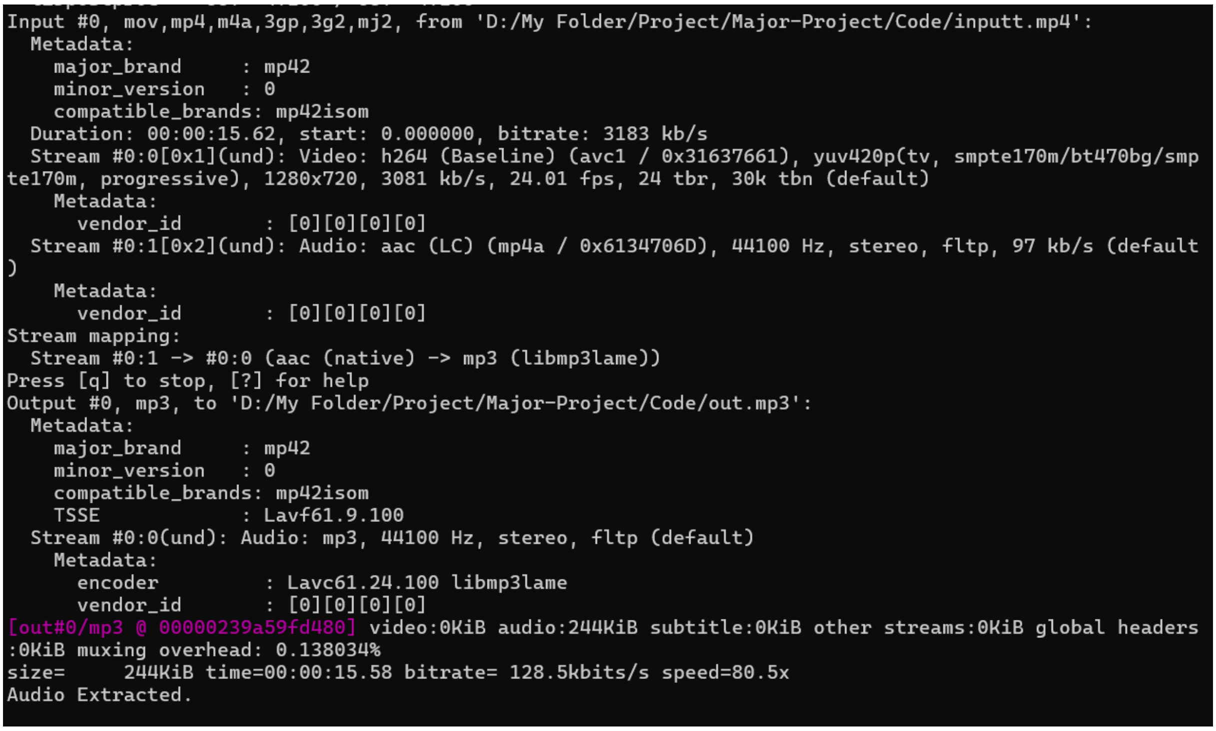
Results and Discussions

In this chapter, we delve into the results and discussions of the implemented automatic video translator. The following sections provide an overview of the results achieved.

6.1 Testing

A court conversation scene from the Malayalam movie '2020' was primarily used for testing, as it contains sufficient dialogue segments and includes both male and female characters. The results in the following section are based on this scene.

6.1.1 Audio Extraction



```
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'D:/My Folder/Project/Major-Project/Code/inputt.mp4':
  Metadata:
    major_brand     : mp42
    minor_version   : 0
    compatible_brands: mp42isom
  Duration: 00:00:15.62, start: 0.000000, bitrate: 3183 kb/s
  Stream #0:0[0x1](und): Video: h264 (Baseline) (avc1 / 0x31637661), yuv420p(tv, smpte170m/bt470bg/smp
te170m, progressive), 1280x720, 3081 kb/s, 24.01 fps, 24 tbr, 30k tbn (default)
    Metadata:
      vendor_id       : [0][0][0][0]
  Stream #0:1[0x2](und): Audio: aac (LC) (mp4a / 0x6134706D), 44100 Hz, stereo, fltp, 97 kb/s (default
)
    Metadata:
      vendor_id       : [0][0][0][0]
Stream mapping:
  Stream #0:1 -> #0:0 (aac (native) -> mp3 (libmp3lame))
Press [q] to stop, [?] for help
Output #0, mp3, to 'D:/My Folder/Project/Major-Project/Code/out.mp3':
  Metadata:
    major_brand     : mp42
    minor_version   : 0
    compatible_brands: mp42isom
    TSSE            : Lavf61.9.100
  Stream #0:0(und): Audio: mp3, 44100 Hz, stereo, fltp (default)
    Metadata:
      encoder         : Lavc61.24.100 libmp3lame
      vendor_id       : [0][0][0][0]
[out#0/mp3 @ 00000239a59fd480] video:0KiB audio:244KiB subtitle:0KiB other streams:0KiB global headers
:0KiB muxing overhead: 0.138034%
size=     244KiB time=00:00:15.58 bitrate= 128.5kbits/s speed=80.5x
Audio Extracted.
```

Figure 6.1: Audio is extracted from the input video file and saved as out.mp3

6.1.2 Audio Separation

The audio was separated into vocals and accompaniments using Spleeter.

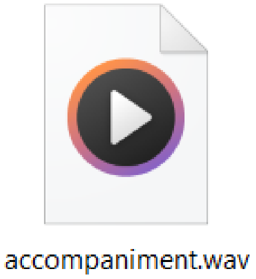


Figure 6.2: Audio was separated into vocals and accompaniment

6.1.3 Audio Segmentation

Timestamps of speech segments were detected, and the audio was trimmed into smaller dialogue segments.

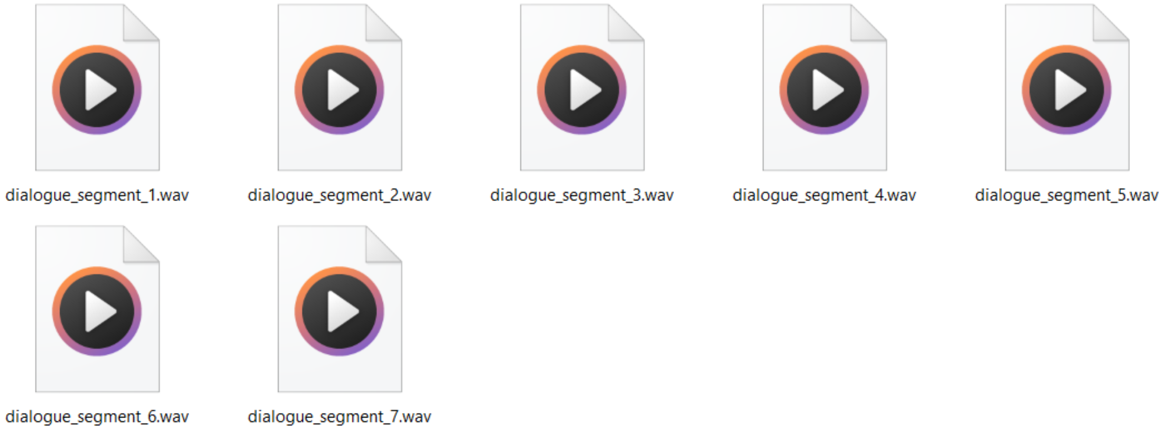
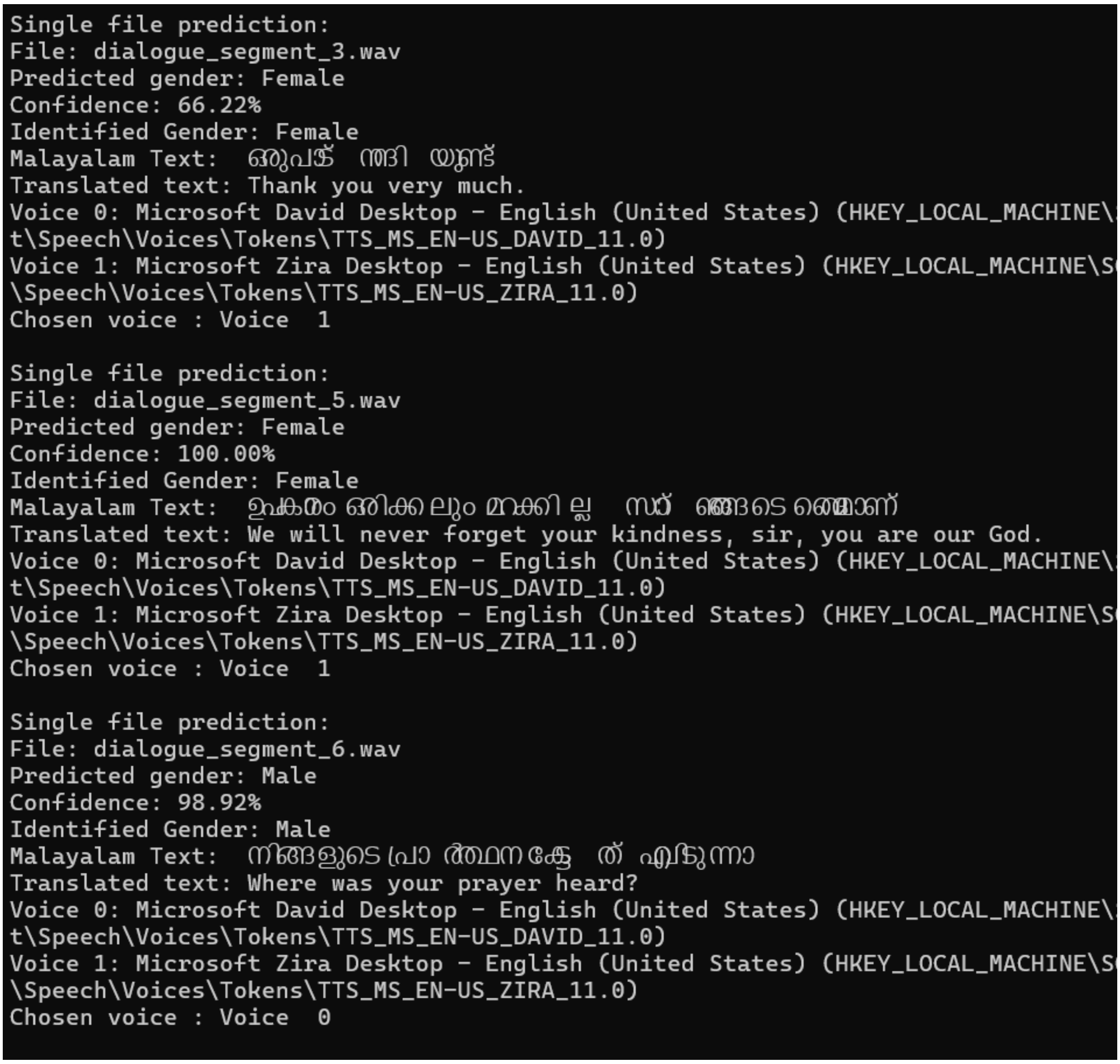


Figure 6.3: Vocals were Segmented and stored as small dialogue segments

6.1.4 Malayalam Audio to English Audio Conversion

The Malayalam text and the speaker's gender are recognized from the dialogue segments. The text is then translated into English and the translated English text is converted to speech using a suitable voice.



```
Single file prediction:
File: dialogue_segment_3.wav
Predicted gender: Female
Confidence: 66.22%
Identified Gender: Female
Malayalam Text:  ഒരുപാട് നന്ദി യുണ്ട്
Translated text: Thank you very much.
Voice 0: Microsoft David Desktop - English (United States) (HKEY_LOCAL_MACHINE\
t\Speech\Voices\Tokens\TTS_MS_EN-US_DAVID_11.0)
Voice 1: Microsoft Zira Desktop - English (United States) (HKEY_LOCAL_MACHINE\S
\Speech\Voices\Tokens\TTS_MS_EN-US_ZIRA_11.0)
Chosen voice : Voice  1

Single file prediction:
File: dialogue_segment_5.wav
Predicted gender: Female
Confidence: 100.00%
Identified Gender: Female
Malayalam Text:  ഉപകാരം ഒരിക്കലും മറക്കില്ല  സാറ് ഞങ്ങടെ ദൈവമാണ്
Translated text: We will never forget your kindness, sir, you are our God.
Voice 0: Microsoft David Desktop - English (United States) (HKEY_LOCAL_MACHINE\
t\Speech\Voices\Tokens\TTS_MS_EN-US_DAVID_11.0)
Voice 1: Microsoft Zira Desktop - English (United States) (HKEY_LOCAL_MACHINE\S
\Speech\Voices\Tokens\TTS_MS_EN-US_ZIRA_11.0)
Chosen voice : Voice  1

Single file prediction:
File: dialogue_segment_6.wav
Predicted gender: Male
Confidence: 98.92%
Identified Gender: Male
Malayalam Text:  നിങ്ങളുടെ പ്രാർത്ഥന കേട്ടത് എവിടുന്നാ
Translated text: Where was your prayer heard?
Voice 0: Microsoft David Desktop - English (United States) (HKEY_LOCAL_MACHINE\
t\Speech\Voices\Tokens\TTS_MS_EN-US_DAVID_11.0)
Voice 1: Microsoft Zira Desktop - English (United States) (HKEY_LOCAL_MACHINE\S
\Speech\Voices\Tokens\TTS_MS_EN-US_ZIRA_11.0)
Chosen voice : Voice  0
```

Figure 6.4: Steps involved in converting Malayalam audio to English audio.

6.1.5 Final Output

The output video file was generated by combining the English dialogues and accompaniments with the original video.

Figure 6.5: Final Output video was generated.

6.2 Quantitative Results

6.2.1 Model

The Model Learning Curves and Confusion matrix of the custom ResNet-based CNN for audio classification are given below.

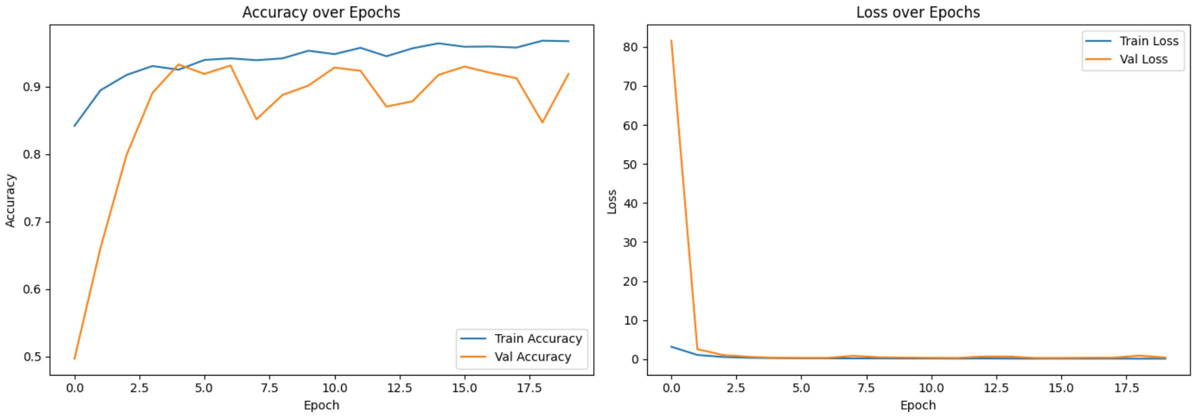


Figure 6.6: Training and Validation Accuracy and Loss Graph.

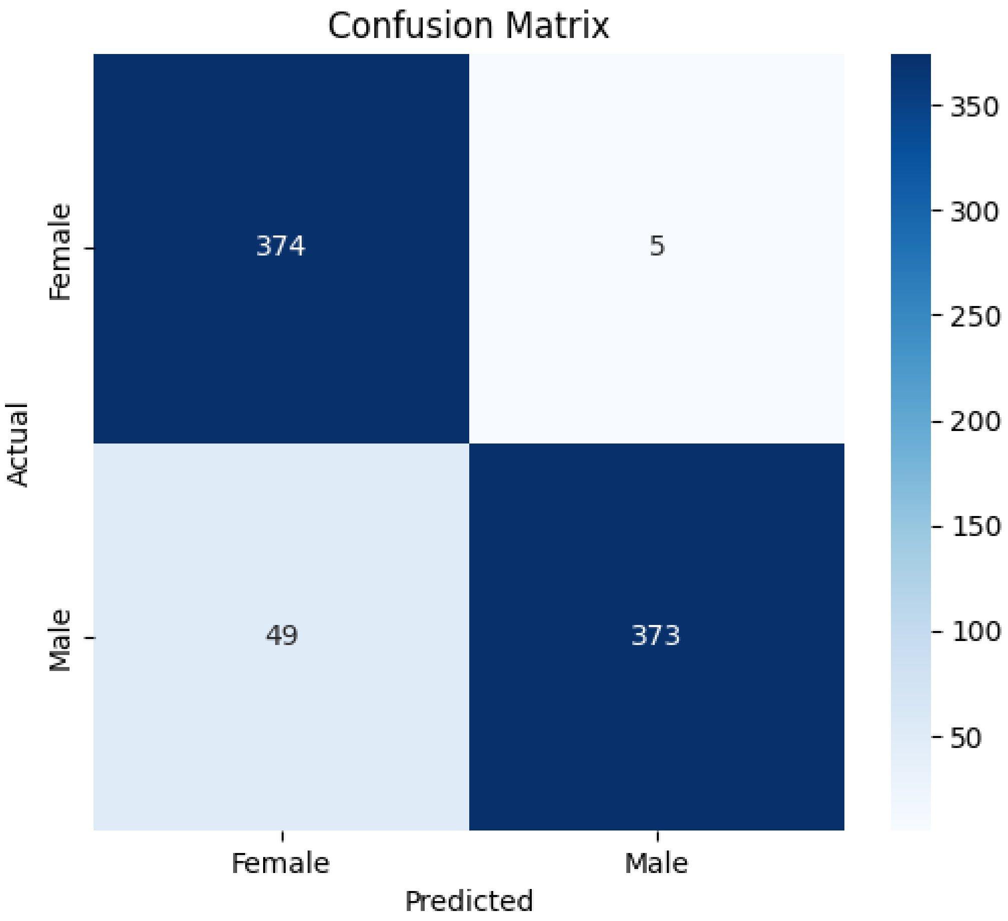


Figure 6.7: Confusion matrix based on validation.

The figures show that the loss is very low and the model achieves over 90% accuracy by the end of 20 epochs.

6.2.2 Execution

The durations of various movie scenes and their corresponding execution times for generating outputs were recorded in the table below.

Movie Name	Duration	Dialogue Segments	Execution Time
2020	15 sec	7	1.08 min
Kaduva	52 sec	26	1.42 min
Hello Mummy	1.29 min	39	2.39 min

Table 6.1: Movie Scene Durations and Execution Times

6.2.3 Discussion

A few observations can be made from the outputs. In some input videos, parts of the audio may not be identified either due to extreme noise or due to low volume. Also if there happens to be overlapping voices in the input video, chances of it processing further is quite low. In some rare cases, male and female voices may not be identified accurately. Sometimes, during the process of converting audio to English, the synchronization of background music and merging with the original video according to timestamps that were detected earlier can result in the final audio being played unnaturally slowly or quickly in some sections, in order to match with the original timing.

But overall we can observe that the generated output video satisfies the intended objectives to a great extent.

6.3 Chapter Conclusion

This chapter provided a thorough analysis of the Automated Video Translator system in terms of its performance in important modules. Testing was performed on a Malayalam movie scene, which showed the ability of the system to extract, preprocess, and segment audio correctly, and also translate Malayalam speech into English text and synthesized audio. The gender identification module scored high accuracy (95%), guaranteeing speaker authenticity on the translated output. Quantitative outcomes showed effective execution times with shorter video segments being processed within less than 2 minutes. However, challenges such as overlapping voices, low-volume audio, and occasional synchronization issues were identified. Despite these limitations, the system successfully met its core objectives thus presenting an adaptable solution for facilitating language translation across Malayalam content.

Chapter 7

Conclusion and Future Scope

To conclude, the "Automated Video Translator" is a solution to the challenge of translating Malayalam audio into English, emphasizing accuracy, efficiency, and accessibility. Its modular design integrates a robust system architecture that includes audio extraction, preprocessing, speech-to-text conversion, translation, text-to-speech synthesis, synchronization, and final merging. Each module plays a critical role in maintaining the accuracy, timing, and natural feel of the translated content. To enhance the system's robustness and applicability, the following extensions are proposed:

1. **Support for Multiple Languages**

   The tool can be upgraded to support other regional languages and become a multilingual tool for translation.

2. **Real-Time Translation**

   The tool, with more development, can provide real-time transcription and translation for live events, webinars, or broadcasts.

By leveraging advanced technologies and custom models, the system transforms Malayalam video content into globally accessible media thus enhancing the reach of regional content in an increasingly interconnected world. This implementation lays a strong foundation for building more advanced, multilingual, and real-time translation systems in the future.

References

[1] Y. Bai, J. Yi, J. Tao, Z. Tian, Z. Wen, and S. Zhang, “Listen attentively, and spell once: Whole sentence generation via a non-autoregressive architecture for low-latency speech recognition,” in *Conference of the International Speech Communication Association (INTERSPEECH)*, Oct 2020.

[2] X. Y. L. L. C. M. Duan, C. and M. Zhang, “Modeling future cost for neural machine translation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 770–781, Jan 2021.

[3] D. Serdyuk, O. Braga, and O. Siohan, “Audio-visual speech recognition is worth 32 32 8 voxels,” *arXiv preprint arXiv:2109.09536*, 2021.

[4] A. Ali, M. Magdy, M. Alfawzy, M. Ghaly, and H. Abbas, “Arabic speech synthesis using deep neural networks,” in *International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, 2021, pp. 1–6.

[5] R. Hsiao, D. Can, T. Ng, R. Travadi, and A. Ghoshal, “Online automatic speech recognition with listen, attend and spell model,” *IEEE Signal Processing Letters*, vol. 27, pp. 1889–1893, 2020.

[6] A. H. P., J. Kunjumon, S. R., and A. S. Ansalem, “Malayalam speech to text conversion using deep learning,” *IOSR Journal of Engineering (IOSRJEN)*, vol. 11, no. 7, Series-II, pp. 24–30, 2021, available online at https://www.iosrjen.org.

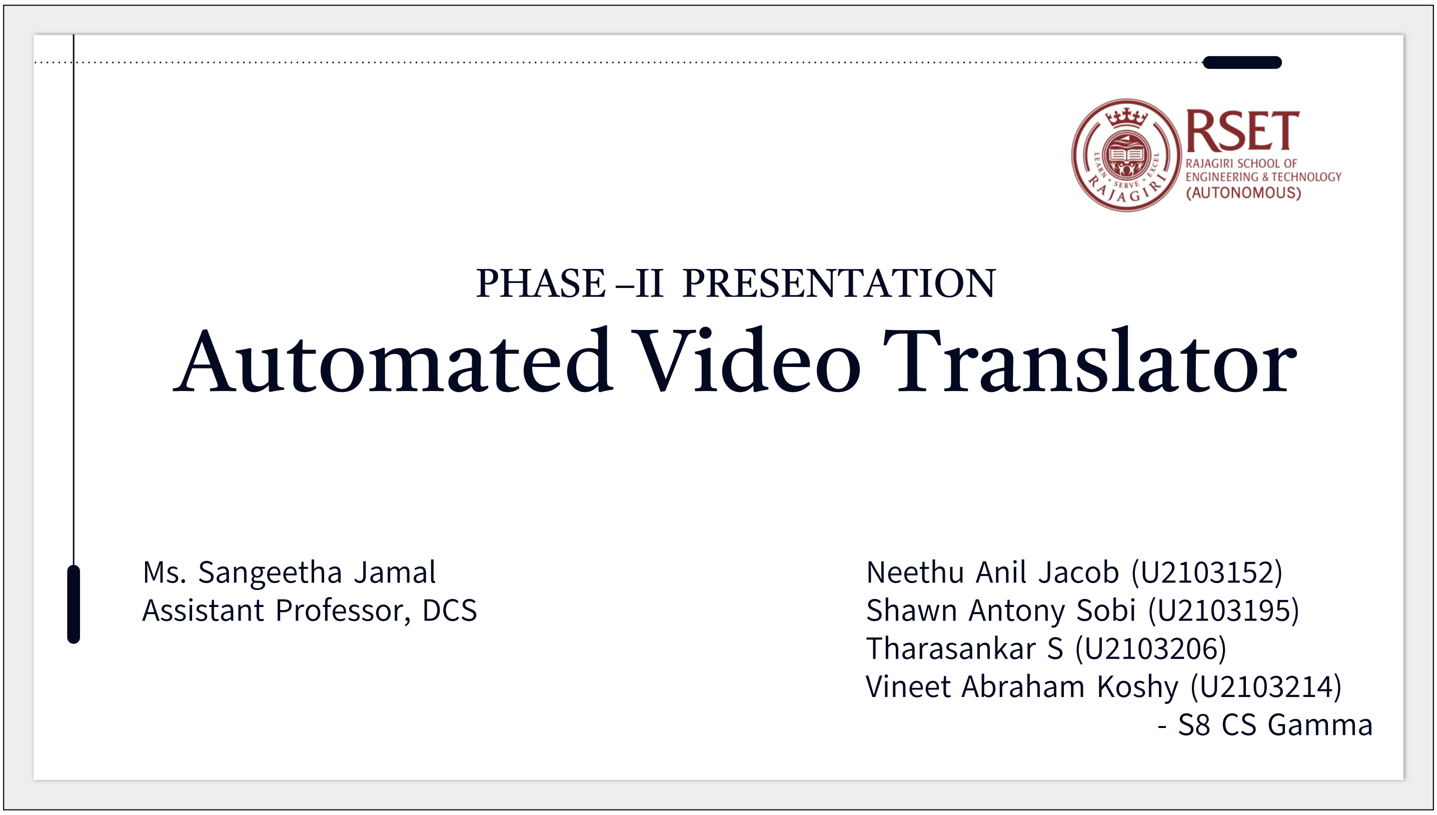
[7] K. Akshay, A. Das, C. Vincent, B. Babu, and P. Rasmi, “Real time translation of malayalam notice boards to english directions,” *International Journal of Computer Applications*, vol. 178, no. 26, pp. 6–10, Jun 2019.

[8] Y.-C. Fan, Y. Qian, F. Xie, and F. K. Soong, “Tts synthesis with bidirectional lstm based recurrent neural networks,” in *Proceedings of Interspeech.* Singapore: ISCA, Sep 2014.

[9] N. B and S. Joseph, “A hybrid approach to english to malayalam machine translation,” *International Journal of Computer Applications*, vol. 81, no. 8, pp. 11–15, Nov 2013.

[10] P. Kumar and H. S. Jayanna, “Development of speaker-independent automatic speech recognition system for kannada language,” *Indian Journal of Science and Technology*, vol. 15, no. 8, pp. 333–342, Feb 2022.

Appendix A: Presentation

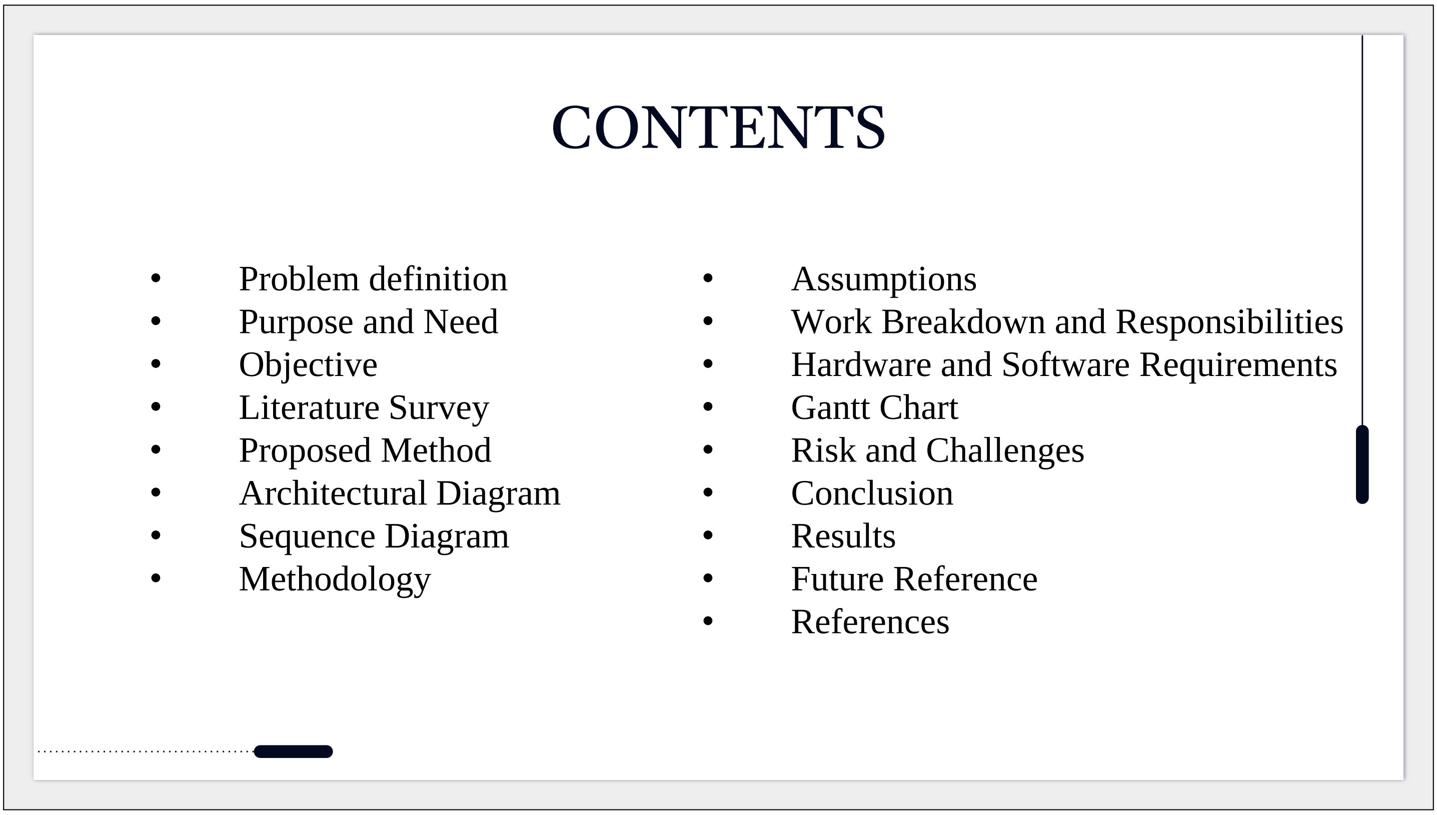


PHASE –II PRESENTATION

Automated Video Translator

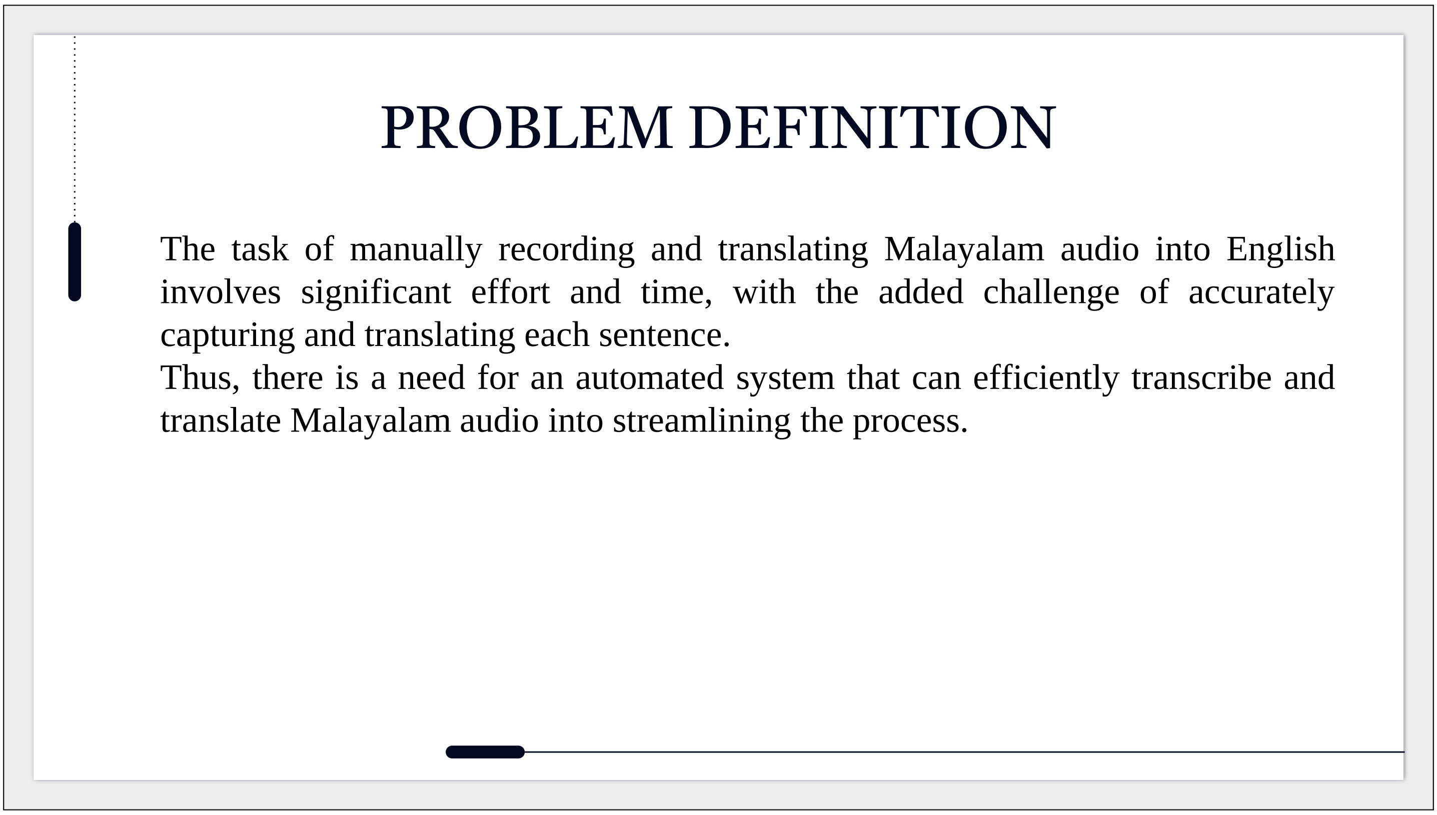
Ms. Sangeetha Jamal
Assistant Professor, DCS

Neethu Anil Jacob (U2103152)
Shawn Antony Sobi (U2103195)
Tharasankar S (U2103206)
Vineet Abraham Koshy (U2103214)
- S8 CS Gamma



CONTENTS

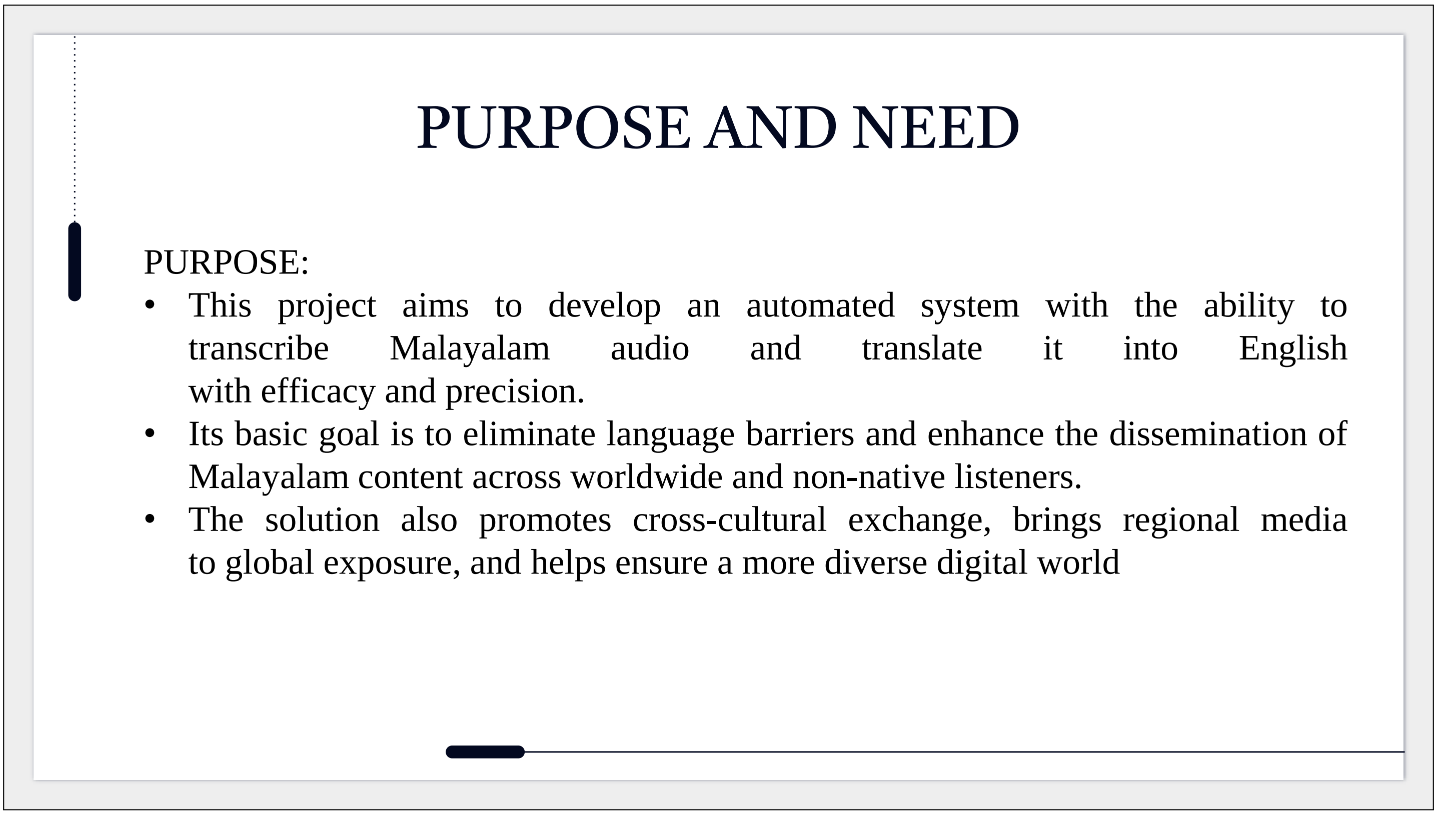
- Problem definition
- Purpose and Need
- Objective
- Literature Survey
- Proposed Method
- Architectural Diagram
- Sequence Diagram
- Methodology
- Assumptions
- Work Breakdown and Responsibilities
- Hardware and Software Requirements
- Gantt Chart
- Risk and Challenges
- Conclusion
- Results
- Future Reference
- References



PROBLEM DEFINITION

The task of manually recording and translating Malayalam audio into English involves significant effort and time, with the added challenge of accurately capturing and translating each sentence.

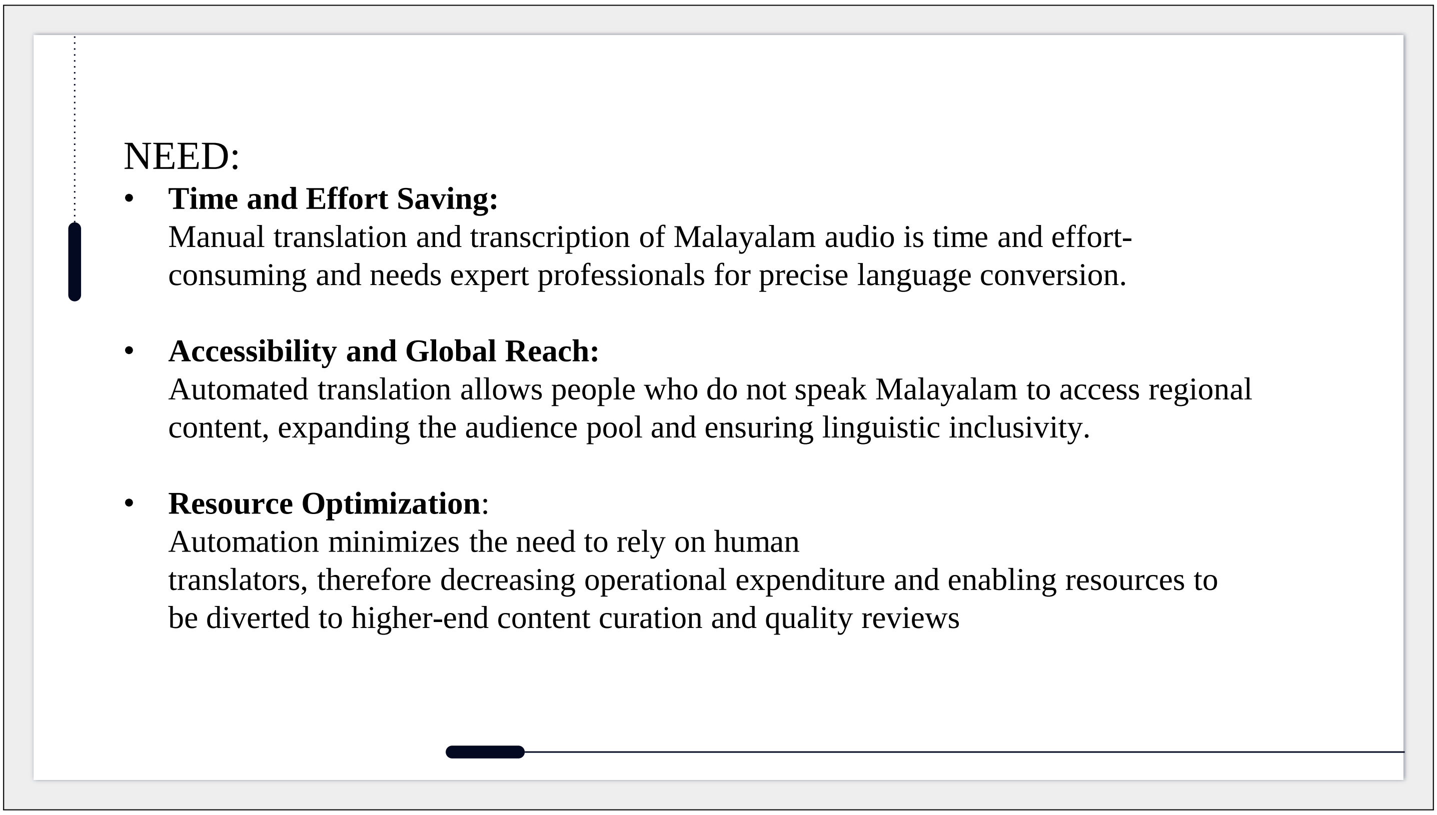
Thus, there is a need for an automated system that can efficiently transcribe and translate Malayalam audio into streamlining the process.



PURPOSE AND NEED

PURPOSE:

- This project aims to develop an automated system with the ability to transcribe Malayalam audio and translate it into English with efficacy and precision.
- Its basic goal is to eliminate language barriers and enhance the dissemination of Malayalam content across worldwide and non-native listeners.
- The solution also promotes cross-cultural exchange, brings regional media to global exposure, and helps ensure a more diverse digital world

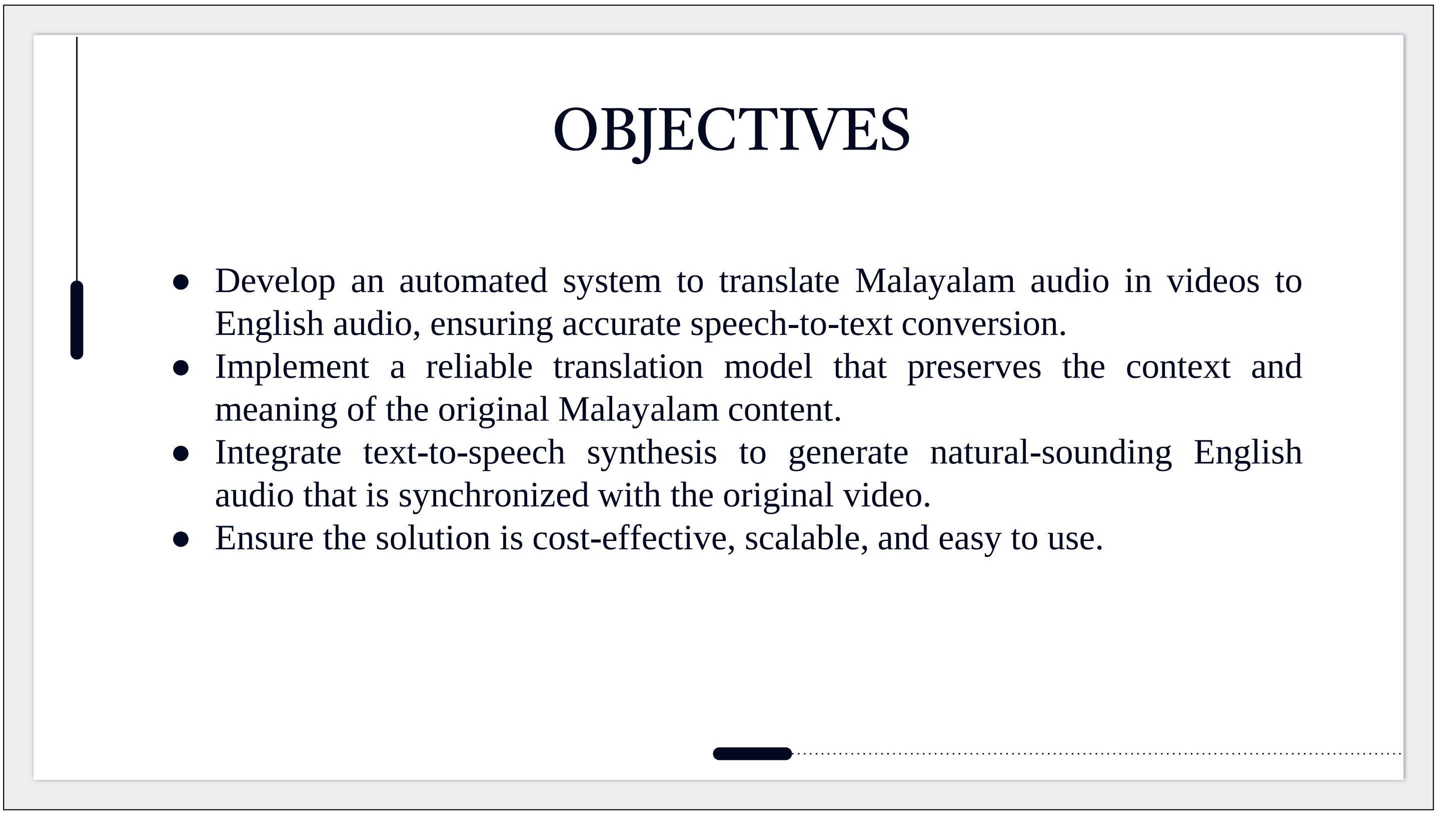


NEED:

- **Time and Effort Saving:**
  Manual translation and transcription of Malayalam audio is time and effort-consuming and needs expert professionals for precise language conversion.

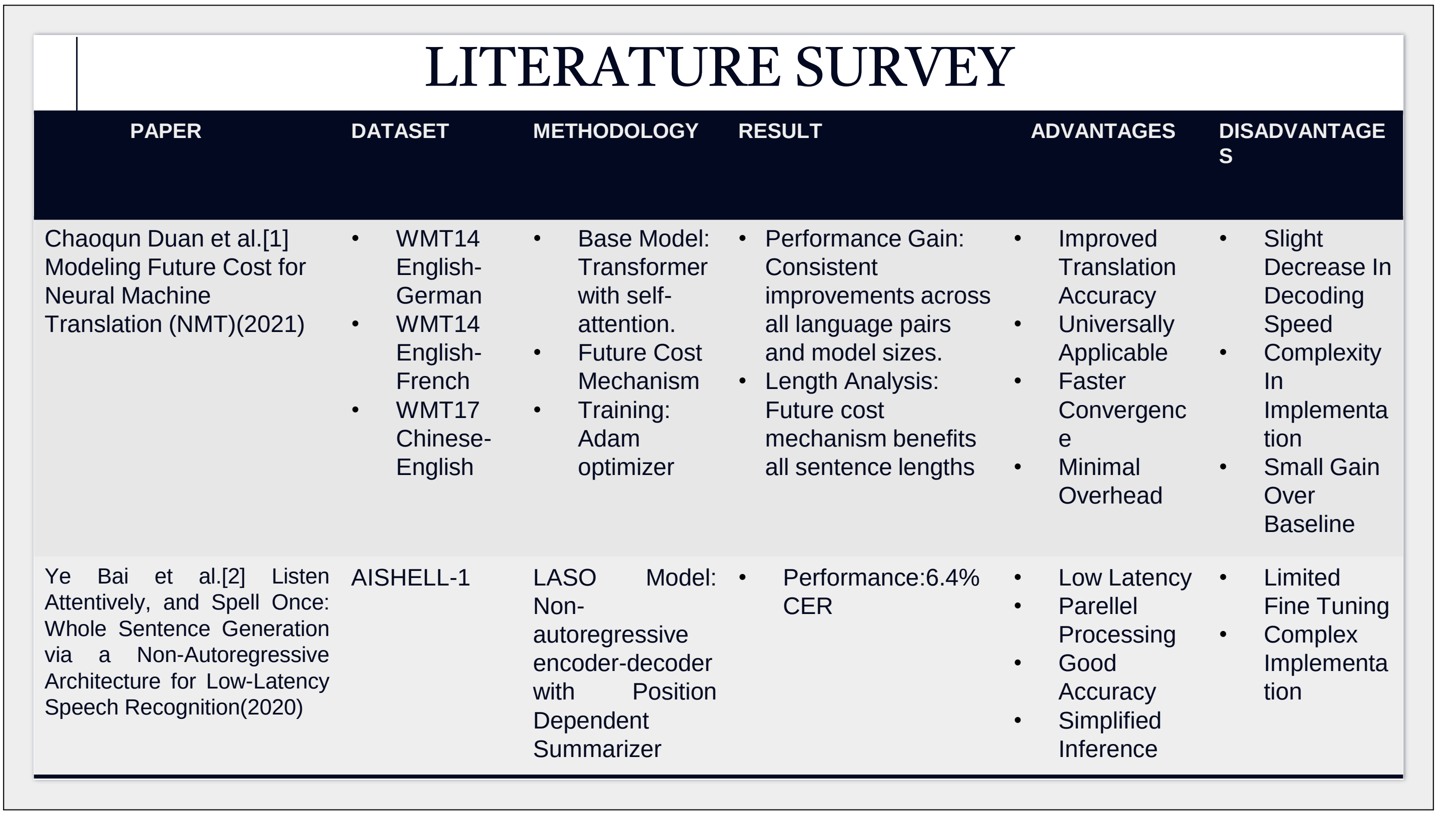
- **Accessibility and Global Reach:**
  Automated translation allows people who do not speak Malayalam to access regional content, expanding the audience pool and ensuring linguistic inclusivity.

- **Resource Optimization**:
  Automation minimizes the need to rely on human translators, therefore decreasing operational expenditure and enabling resources to be diverted to higher-end content curation and quality reviews



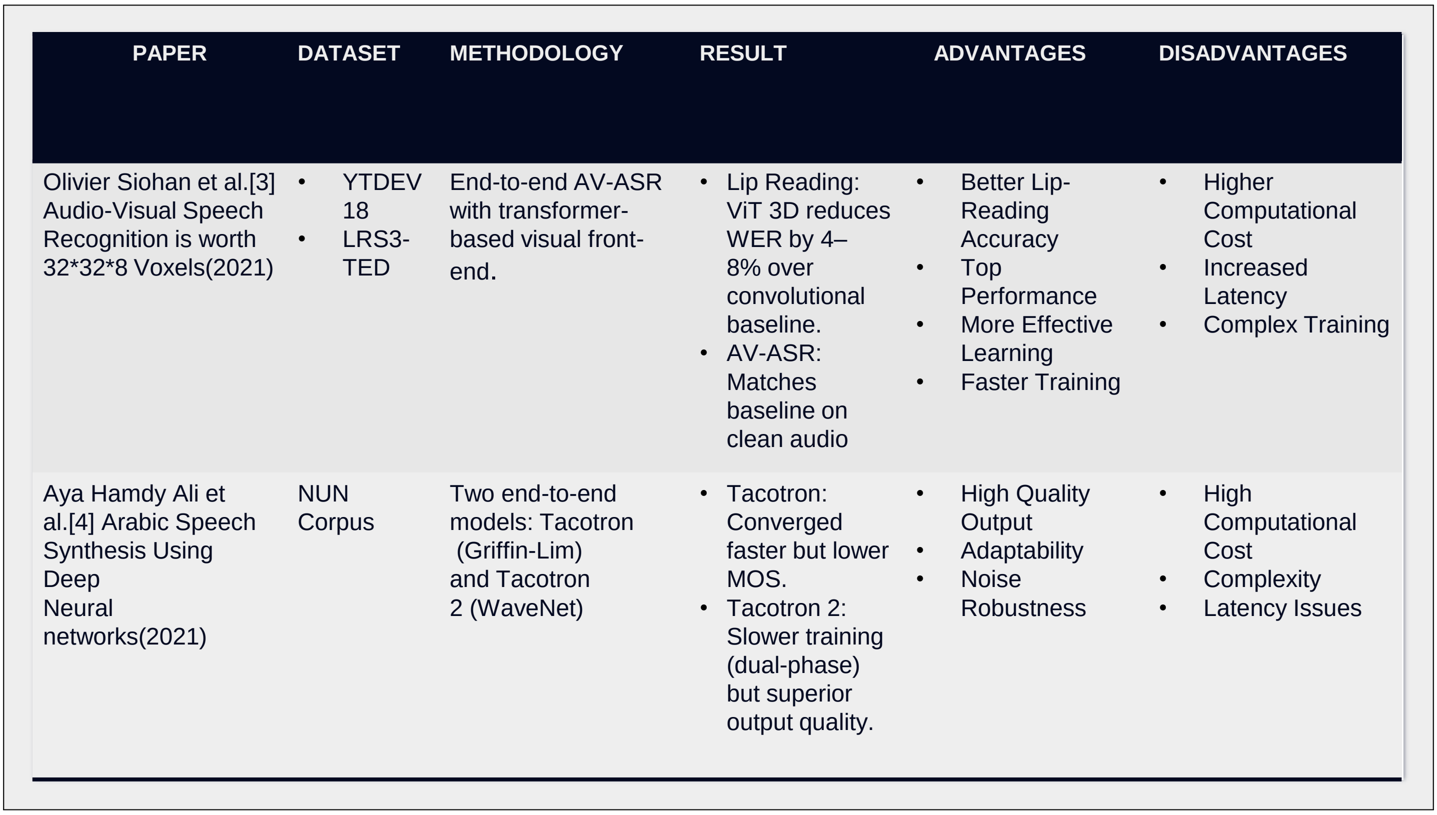
OBJECTIVES

- Develop an automated system to translate Malayalam audio in videos to English audio, ensuring accurate speech-to-text conversion.
- Implement a reliable translation model that preserves the context and meaning of the original Malayalam content.
- Integrate text-to-speech synthesis to generate natural-sounding English audio that is synchronized with the original video.
- Ensure the solution is cost-effective, scalable, and easy to use.

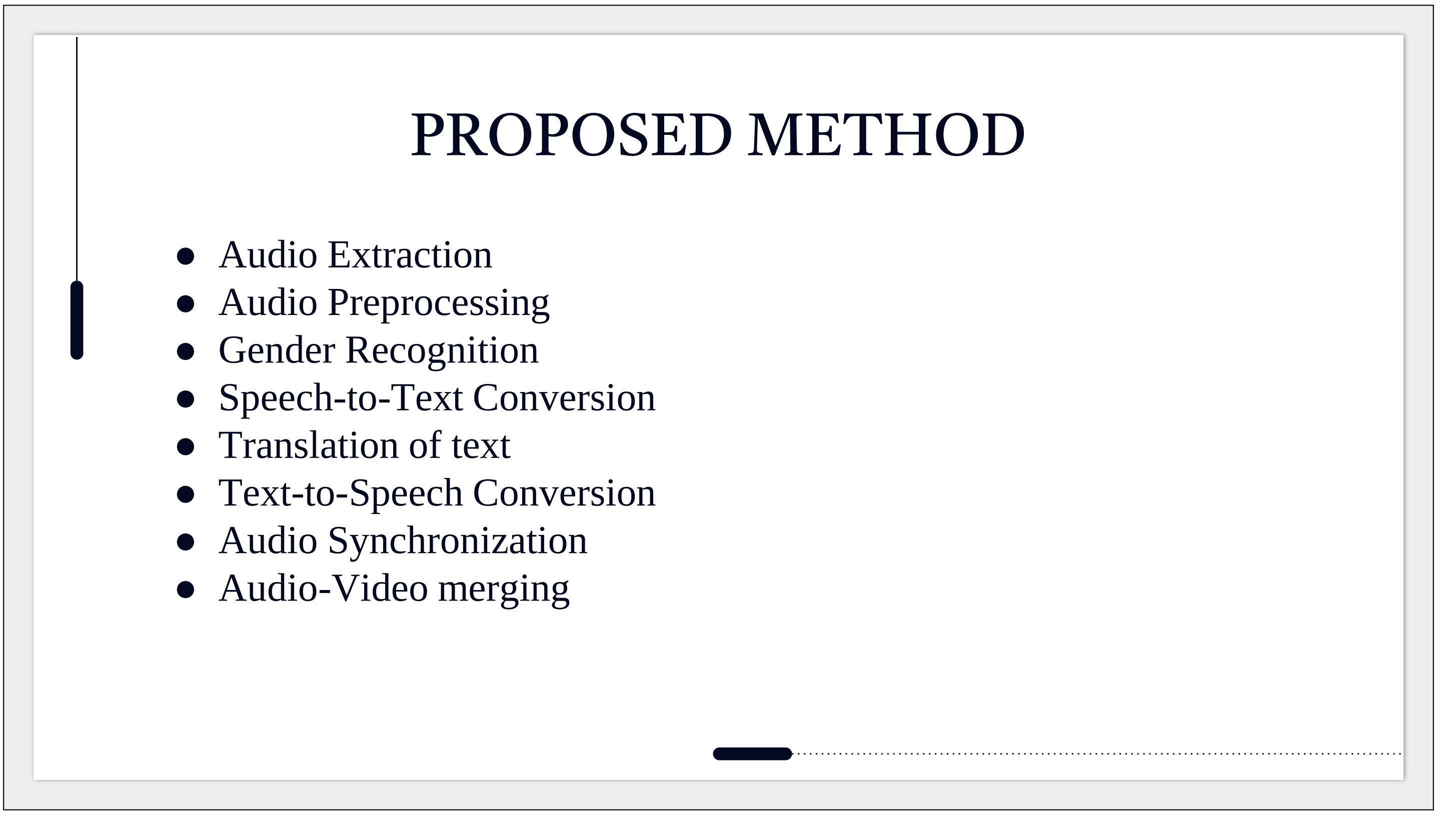


LITERATURE SURVEY

PAPER	DATASET	METHODOLOGY	RESULT	ADVANTAGES	DISADVANTAGES
Chaoqun Duan et al.[1] Modeling Future Cost for Neural Machine Translation (NMT)(2021)	• WMT14 English-German • WMT14 English-French • WMT17 Chinese-English	• Base Model: Transformer with self-attention. • Future Cost Mechanism • Training: Adam optimizer	• Performance Gain: Consistent improvements across all language pairs and model sizes. • Length Analysis: Future cost mechanism benefits all sentence lengths	• Improved Translation Accuracy • Universally Applicable • Faster Convergence • Minimal Overhead	• Slight Decrease In Decoding Speed • Complexity In Implementation • Small Gain Over Baseline
Ye Bai et al.[2] Listen Attentively, and Spell Once: Whole Sentence Generation via a Non-Autoregressive Architecture for Low-Latency Speech Recognition(2020)	AISHELL-1	LASO Model: Non-autoregressive encoder-decoder with Position Dependent Summarizer	• Performance:6.4% CER	• Low Latency • Parellel Processing • Good Accuracy • Simplified Inference	• Limited Fine Tuning • Complex Implementation

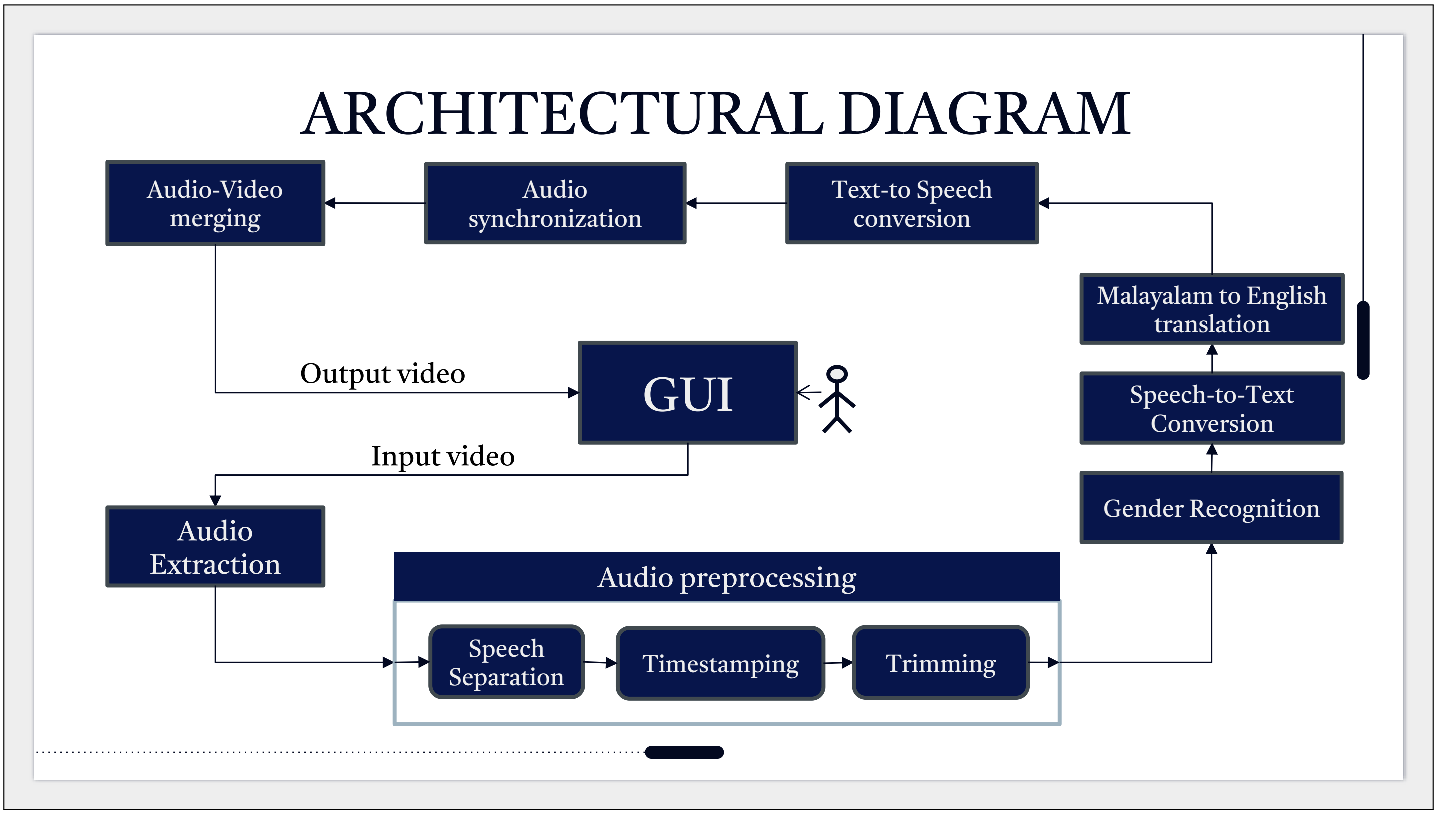


PAPER	DATASET	METHODOLOGY	RESULT	ADVANTAGES	DISADVANTAGES
Olivier Siohan et al.[3] Audio-Visual Speech Recognition is worth 32*32*8 Voxels(2021)	• YTDEV18 • LRS3-TED	End-to-end AV-ASR with transformer-based visual front-end.	• Lip Reading: ViT 3D reduces WER by 4–8% over convolutional baseline. • AV-ASR: Matches baseline on clean audio	• Better Lip-Reading Accuracy • Top Performance • More Effective Learning • Faster Training	• Higher Computational Cost • Increased Latency • Complex Training
Aya Hamdy Ali et al.[4] Arabic Speech Synthesis Using Deep Neural networks(2021)	NUN Corpus	Two end-to-end models: Tacotron (Griffin-Lim) and Tacotron 2 (WaveNet)	• Tacotron: Converged faster but lower MOS. • Tacotron 2: Slower training (dual-phase) but superior output quality.	• High Quality Output • Adaptability • Noise Robustness	• High Computational Cost • Complexity • Latency Issues



PROPOSED METHOD

- Audio Extraction
- Audio Preprocessing
- Gender Recognition
- Speech-to-Text Conversion
- Translation of text
- Text-to-Speech Conversion
- Audio Synchronization
- Audio-Video merging



ARCHITECTURAL DIAGRAM

Audio-Video merging

Audio synchronization

Text-to Speech conversion

Malayalam to English translation

Output video

GUI

Speech-to-Text Conversion

Input video

Gender Recognition

Audio Extraction

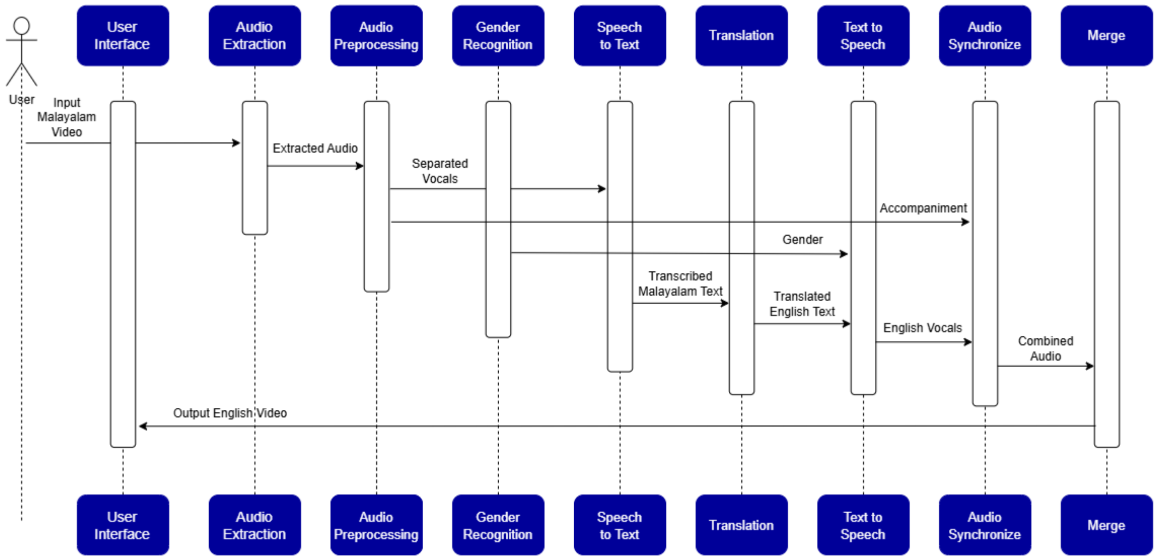
Audio preprocessing

Speech Separation

Timestamping

Trimming

SEQUENCE DIAGRAM



METHODOLOGY

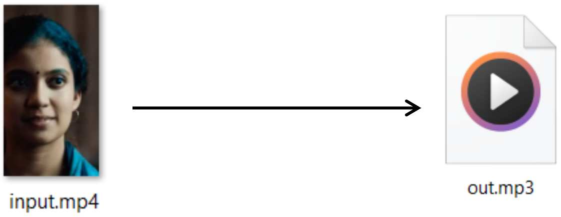
1. Audio Extraction Module

Purpose:

- The audio is extracted from the input video.

Details:

- FFmpeg library is used to extract the audio track from the video file.
- Ensure the extracted audio maintains high quality for further processing.



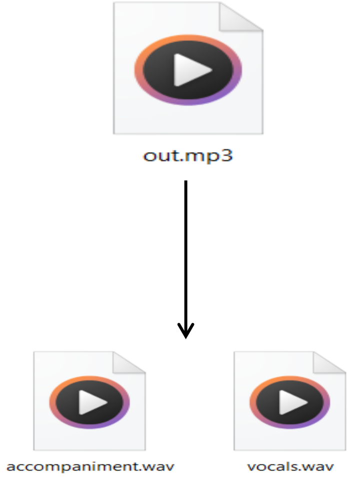
2. Audio Preprocessing Module

Purpose:

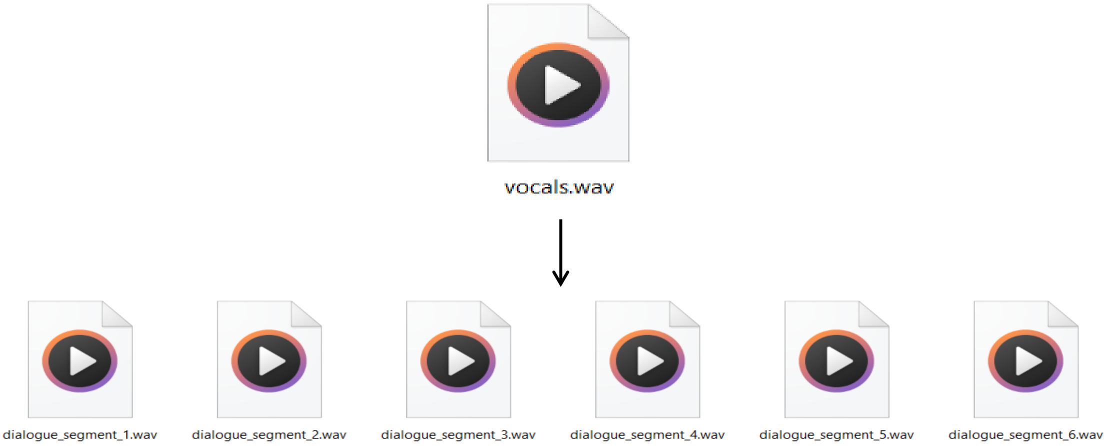
- Separate speech from accompanying sounds, store timestamps of speech segments, and trim the audio accordingly.

Details:

- Spleeter is used to separate speech from background music and noise.



- Timestamps of speech segments are detected using Librosa to identify silent parts and speech boundaries.
- Then the audio is trimmed based on the timestamps using Pydub, creating smaller, manageable clips of just the speech.



3. Gender Recognition

Purpose:

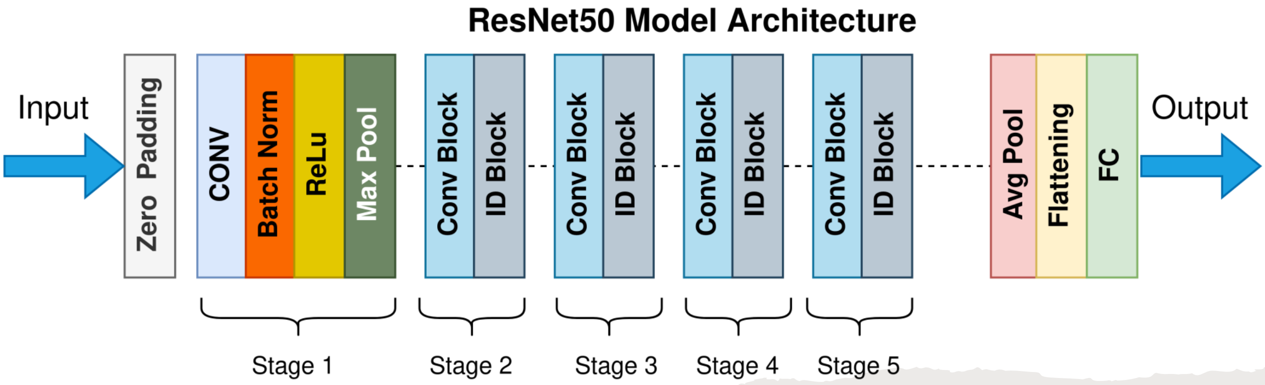
To identify whether the speaker in the audio segment was male or female.

Details:

- A custom ResNet based CNN model was developed to classify the Male and Female voice.
- This was done so as to preserve the speaker authentication in the translated video as well.

ResNet

ResNet stands for Residual Network, a deep convolutional neural network architecture that uses skip connections (or shortcuts) to jump over some layers. This design solves the vanishing gradient problem and makes it possible to train very deep networks effectively.



Difference between Custom ResNet-Based Model and ResNet50/ResNet101

Feature	ResNet50 / ResNet101	Custom ResNet-Based Model
Predefined Architecture	Fixed number of layers (50 or 101)	Flexible layer design
Input Type	Designed for RGB images (224x224x3)	Can be adapted for other inputs (e.g., spectrograms, MFCCs)
Size	Deep (50+ layers)	Can be shallow or deep depending on task
Usage	Generic image classification	Tailored to a specific problem (e.g., speech emotion recognition, gender classification)
Training	Often pretrained on ImageNet	Usually trained from scratch or with domain-specific pretraining

4. Speech-to-Text Conversion Module

Purpose:

- To convert the trimmed Malayalam speech into malayalam text.

Details:

- Speech-to-Text is done using SpeechRecognition library in Python.

5. Translation Module

Purpose:

- To translate the Malayalam text into English text.

Details:

- Google Translate API is used to convert the recognized Malayalam text to English.

6. Text-to-Speech Module

Purpose:

- To convert the translated English text back into speech.

Details:

- Pyttxs3 library is used to generate natural-sounding English audio from the translated English text.

Key Library: pyttsx3.

7. Audio Synchronization Module

Purpose:

- To synchronize the newly generated English audio with the accompanying sounds.

Details:

  - The speech is aligned with the non-speech audio components (like background music) using pydub.

8. Audio-Video Merging Module

Purpose:

- To merge the translated audio back with the original video.

Details:

- MoviePy is used to combine the newly generated English audio with the video.

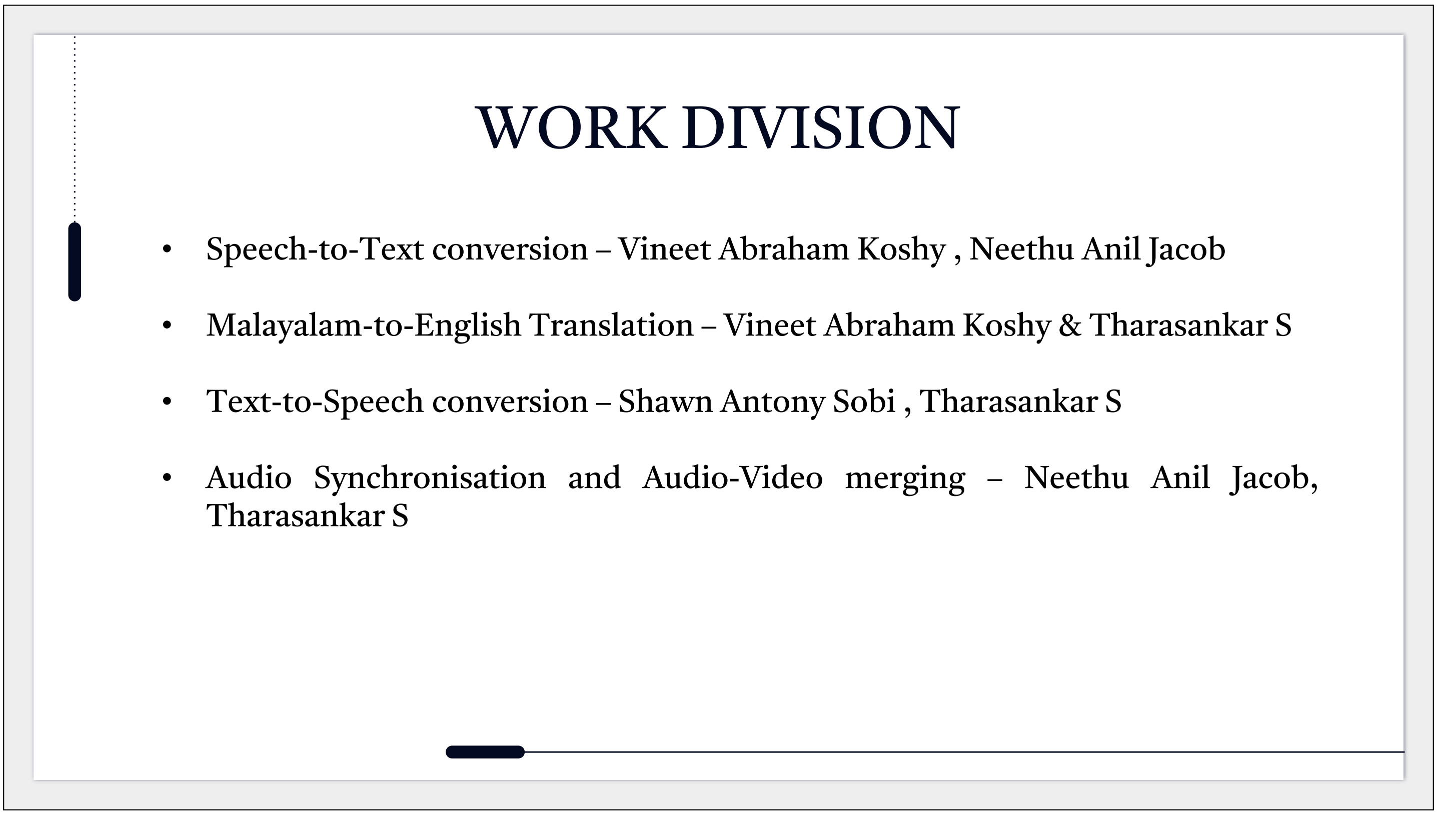
ASSUMPTIONS

•**Clear Audio Input**
The audio provided for transcription is assumed to be of good quality, with minimal background noise and clear speech.

•**Standard Malayalam Dialects**
The system assumes that the audio primarily uses commonly spoken dialects of Malayalam and avoids extremely regional or rare variations.

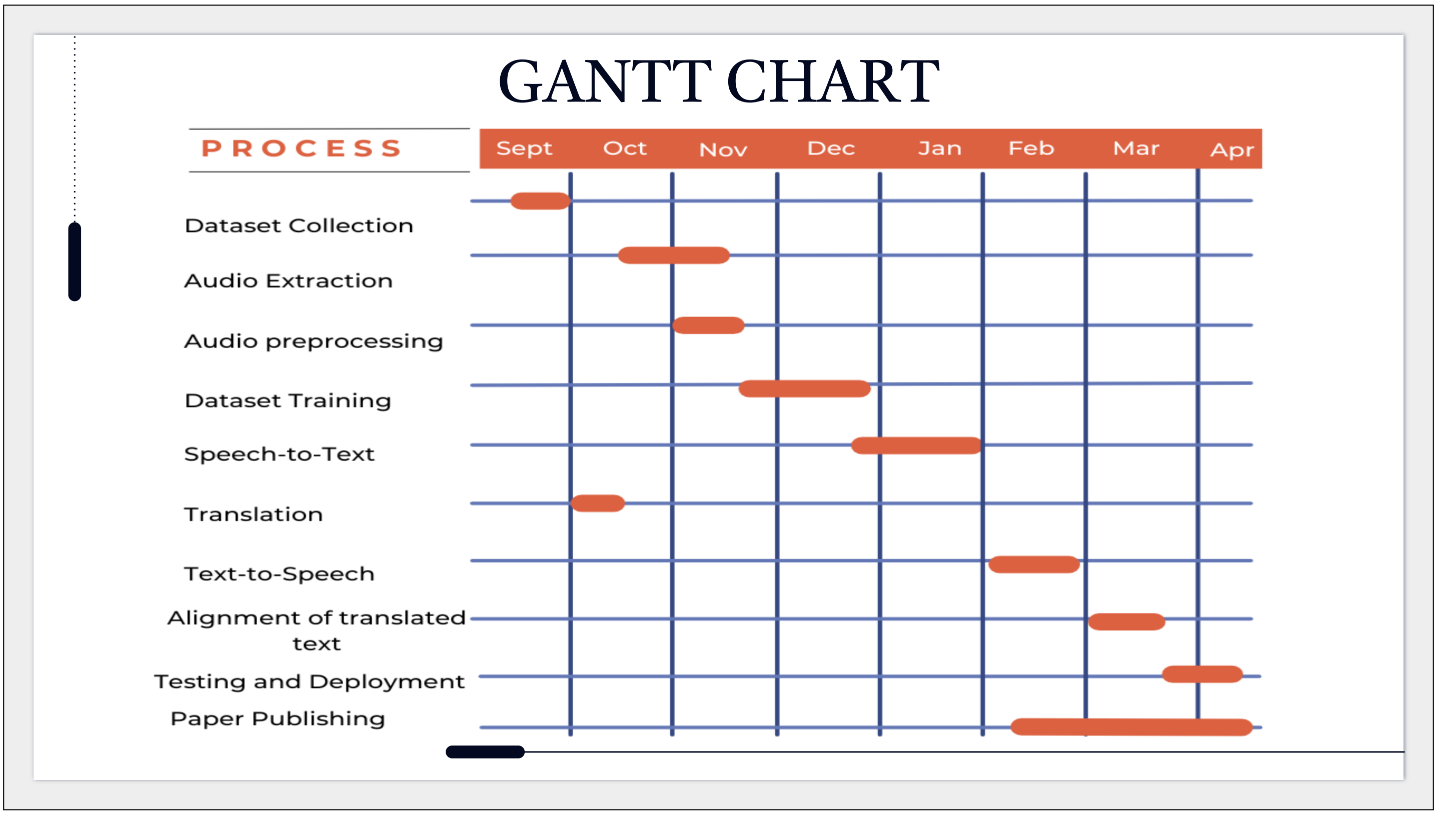
WORK DIVISION

- Audio Extraction – Shawn Antony Sobi & Vineet Abraham Koshy
- Audio Preprocessing – Neethu Anil Jacob, Tharasankar S & Shawn Antony Sobi
- Gender Recognition – Tharasankar S, Shawn Antony Sobi
- Malayalam-to-English Translation – Neethu Anil Jacob, Shawn Antony Sobi



WORK DIVISION

- Speech-to-Text conversion – Vineet Abraham Koshy , Neethu Anil Jacob
- Malayalam-to-English Translation – Vineet Abraham Koshy & Tharasankar S
- Text-to-Speech conversion – Shawn Antony Sobi , Tharasankar S
- Audio Synchronisation and Audio-Video merging – Neethu Anil Jacob, Tharasankar S



GANTT CHART

RISKS AND CHALLENGES

•Speech Recognition Limitations
Variations in dialects, accents, and background noise can affect the accuracy of Malayalam transcription.

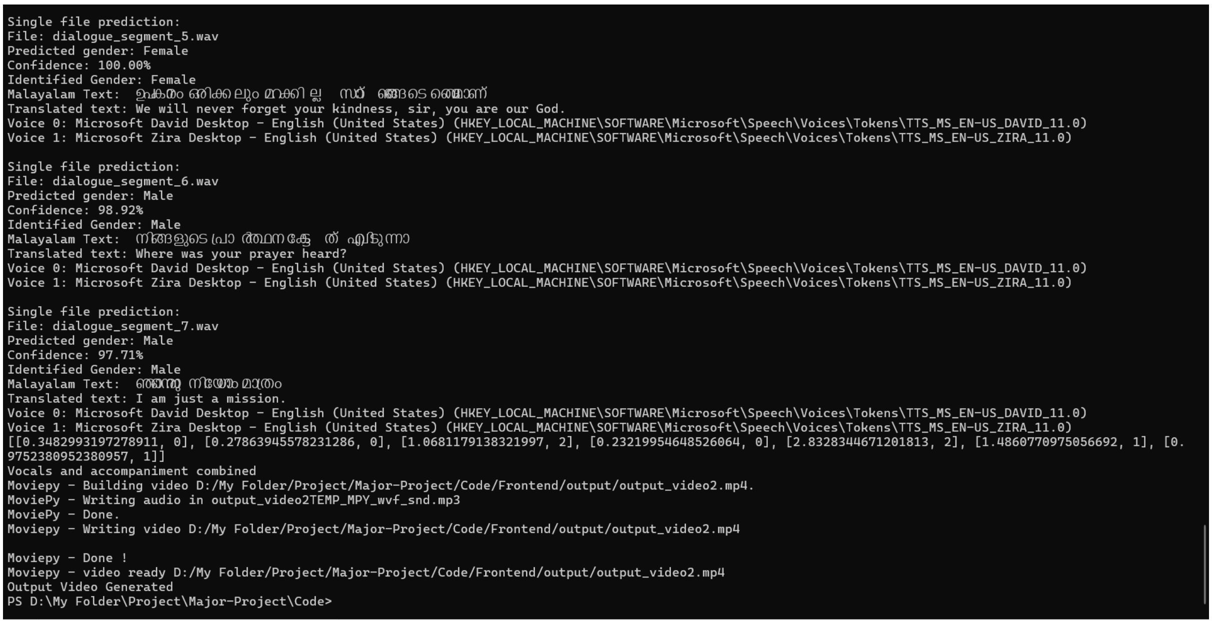
•Translation Inaccuracy
Automated systems may misinterpret idioms, context, or cultural nuances, leading to incorrect or awkward translations.

.

CONCLUSION

The project develops an NLP-based software to automatically translate Malayalam dialogues into English in videos, utilizing advanced speech recognition, translation, and text-to-speech technologies. This model not only streamlines translation and reduces manual effort but also overcomes the limitations that often restrict the global reach of such content.

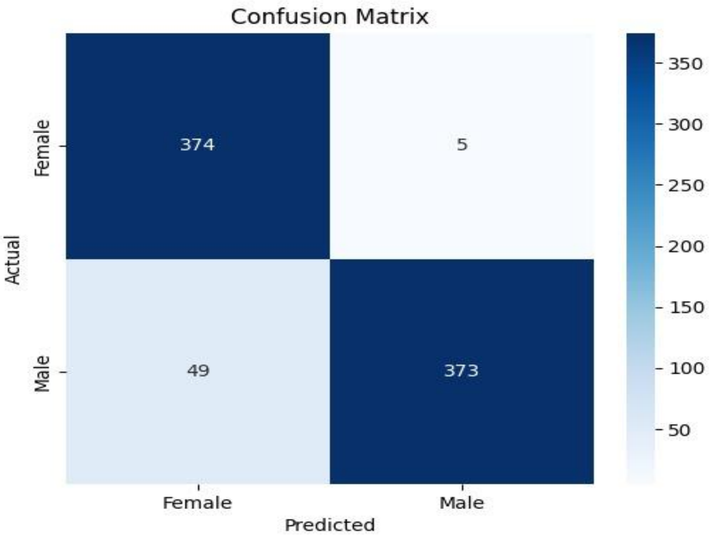
RESULTS



```
Single file prediction:
File: dialogue_segment_5.wav
Predicted gender: Female
Confidence: 100.00%
Identified Gender: Female
Malayalam Text:  ഉപകാരം ഒരിക്കലും മറക്കില്ല  സറ്  ങ്ങടെ ദൈവാണ്
Translated text: We will never forget your kindness, sir, you are our God.
Voice 0: Microsoft David Desktop - English (United States) (HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Voices\Tokens\TTS_MS_EN-US_DAVID_11.0)
Voice 1: Microsoft Zira Desktop - English (United States) (HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Voices\Tokens\TTS_MS_EN-US_ZIRA_11.0)

Single file prediction:
File: dialogue_segment_6.wav
Predicted gender: Male
Confidence: 98.92%
Identified Gender: Male
Malayalam Text:  നിങ്ങളുടെ പ്രാ ർത്ഥന കേ  ത് എവിടുന്നാ
Translated text: Where was your prayer heard?
Voice 0: Microsoft David Desktop - English (United States) (HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Voices\Tokens\TTS_MS_EN-US_DAVID_11.0)
Voice 1: Microsoft Zira Desktop - English (United States) (HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Voices\Tokens\TTS_MS_EN-US_ZIRA_11.0)

Single file prediction:
File: dialogue_segment_7.wav
Predicted gender: Male
Confidence: 97.71%
Identified Gender: Male
Malayalam Text:  ഞാനൊരു നിയോഗം മാത്രം
Translated text: I am just a mission.
Voice 0: Microsoft David Desktop - English (United States) (HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Voices\Tokens\TTS_MS_EN-US_DAVID_11.0)
Voice 1: Microsoft Zira Desktop - English (United States) (HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Voices\Tokens\TTS_MS_EN-US_ZIRA_11.0)
[[0.3482993197278911, 0], [0.27863945578231286, 0], [1.0681179138321997, 2], [0.23219954648526064, 0], [2.8328344671201813, 2], [1.4860770975056692, 1], [0.
9752380952380957, 1]]
Vocals and accompaniment combined
Moviepy - Building video D:/My Folder/Project/Major-Project/Code/Frontend/output/output_video2.mp4.
MoviePy - Writing audio in output_video2TEMP_MPY_wvf_snd.mp3
MoviePy - Done.
Moviepy - Writing video D:/My Folder/Project/Major-Project/Code/Frontend/output/output_video2.mp4

Moviepy - Done !
Moviepy - video ready D:/My Folder/Project/Major-Project/Code/Frontend/output/output_video2.mp4
Output Video Generated
PS D:\My Folder\Project\Major-Project\Code>
```

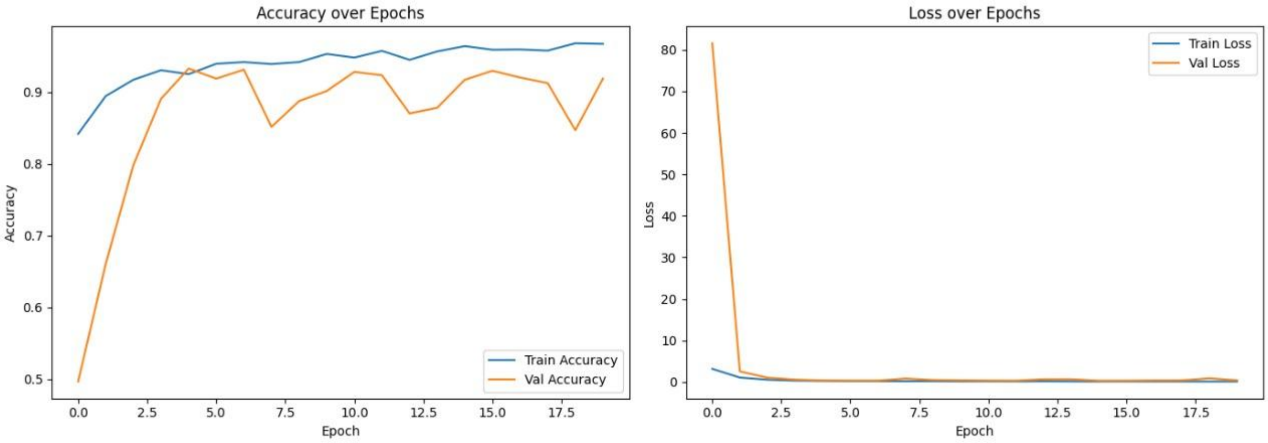
RESULTS

GENDER RECOGNITION:



RESULTS

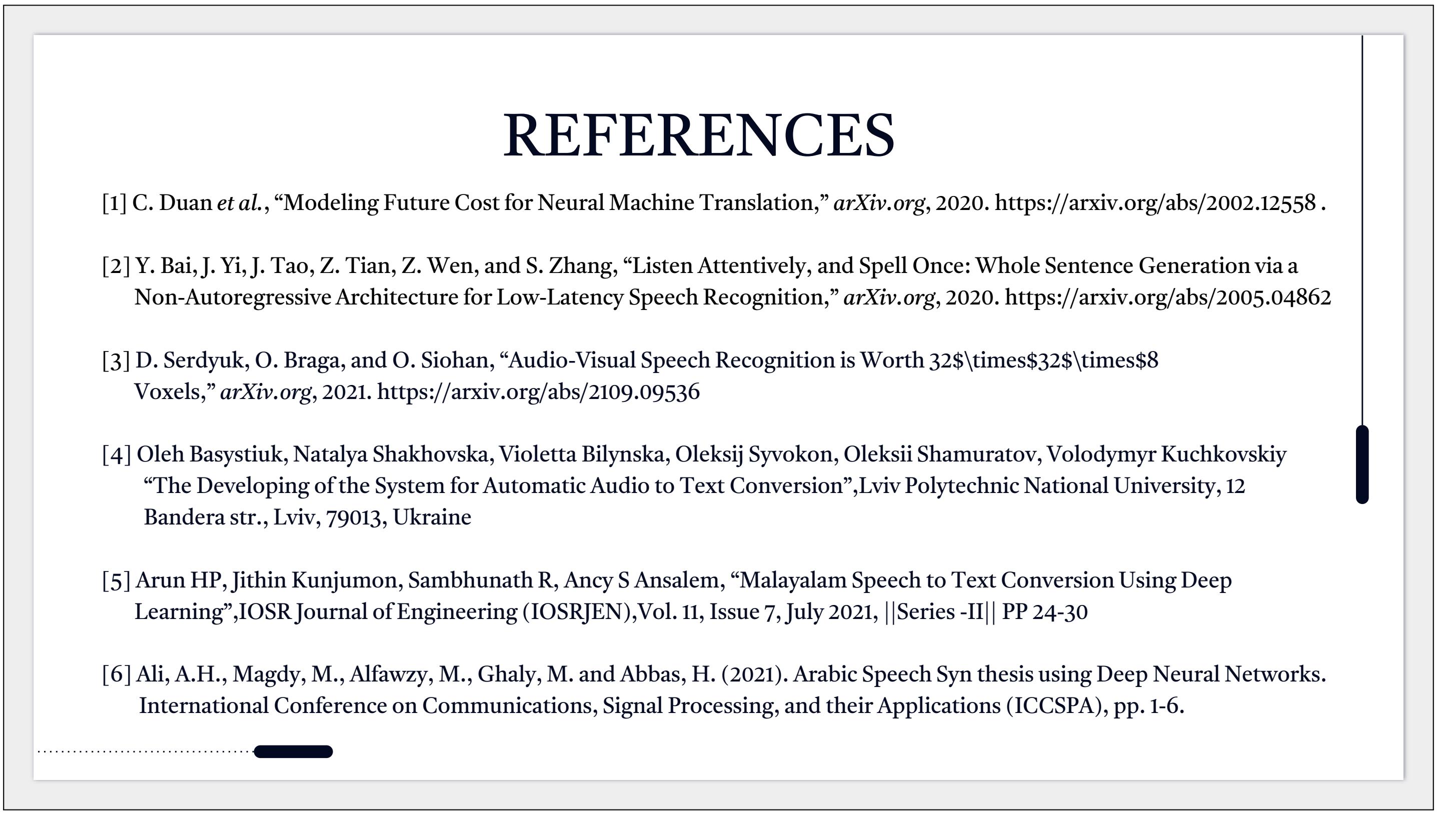
GENDER RECOGNITION:



FUTURE SCOPE

•Support for Multiple Languages
The system can be extended to handle other regional languages, making it a multilingual translation tool.

•Real-Time Translation
With further development, the tool can offer real-time transcription and translation for live events, webinars, or broadcasts



REFERENCES

[1] C. Duan *et al.*, “Modeling Future Cost for Neural Machine Translation,” *arXiv.org*, 2020. https://arxiv.org/abs/2002.12558 .

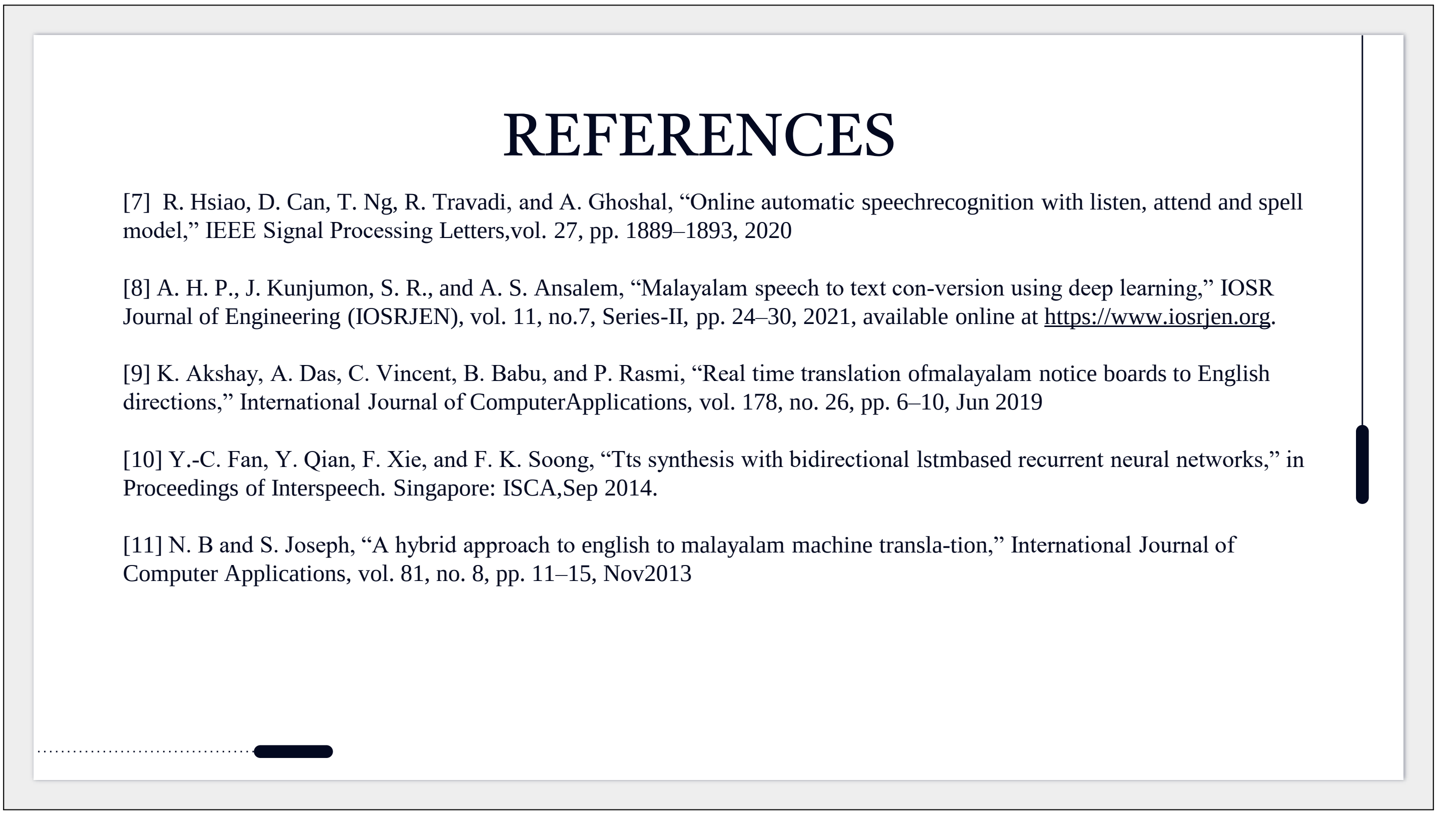
[2] Y. Bai, J. Yi, J. Tao, Z. Tian, Z. Wen, and S. Zhang, “Listen Attentively, and Spell Once: Whole Sentence Generation via a Non-Autoregressive Architecture for Low-Latency Speech Recognition,” *arXiv.org*, 2020. https://arxiv.org/abs/2005.04862

[3] D. Serdyuk, O. Braga, and O. Siohan, “Audio-Visual Speech Recognition is Worth 32$\times$32$\times$8 Voxels,” *arXiv.org*, 2021. https://arxiv.org/abs/2109.09536

[4] Oleh Basystiuk, Natalya Shakhovska, Violetta Bilynska, Oleksij Syvokon, Oleksii Shamuratov, Volodymyr Kuchkovskiy “The Developing of the System for Automatic Audio to Text Conversion”,Lviv Polytechnic National University, 12 Bandera str., Lviv, 79013, Ukraine

[5] Arun HP, Jithin Kunjumon, Sambhunath R, Ancy S Ansalem, “Malayalam Speech to Text Conversion Using Deep Learning”,IOSR Journal of Engineering (IOSRJEN),Vol. 11, Issue 7, July 2021, ||Series -II|| PP 24-30

[6] Ali, A.H., Magdy, M., Alfawzy, M., Ghaly, M. and Abbas, H. (2021). Arabic Speech Syn thesis using Deep Neural Networks. International Conference on Communications, Signal Processing, and their Applications (ICCSPA), pp. 1-6.



REFERENCES

[7] R. Hsiao, D. Can, T. Ng, R. Travadi, and A. Ghoshal, “Online automatic speechrecognition with listen, attend and spell model,” IEEE Signal Processing Letters,vol. 27, pp. 1889–1893, 2020

[8] A. H. P., J. Kunjumon, S. R., and A. S. Ansalem, “Malayalam speech to text con-version using deep learning,” IOSR Journal of Engineering (IOSRJEN), vol. 11, no.7, Series-II, pp. 24–30, 2021, available online at https://www.iosrjen.org.

[9] K. Akshay, A. Das, C. Vincent, B. Babu, and P. Rasmi, “Real time translation ofmalayalam notice boards to English directions,” International Journal of ComputerApplications, vol. 178, no. 26, pp. 6–10, Jun 2019

[10] Y.-C. Fan, Y. Qian, F. Xie, and F. K. Soong, “Tts synthesis with bidirectional lstmbased recurrent neural networks,” in Proceedings of Interspeech. Singapore: ISCA,Sep 2014.

[11] N. B and S. Joseph, “A hybrid approach to english to malayalam machine transla-tion,” International Journal of Computer Applications, vol. 81, no. 8, pp. 11–15, Nov2013

Thank you

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

Vision, Mission, Programme Outcomes and Course Outcomes

Institute Vision

To evolve into a premier technological institution, moulding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

Institute Mission

To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

Department Vision

To become a centre of excellence in Computer Science and Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

Department Mission

To inspire and nurture students, with up-to-date knowledge in Computer Science and Engineering, ethics, team spirit, leadership abilities, innovation and creativity to come out with solutions meeting societal needs.

Programme Outcomes (PO)

Engineering Graduates will be able to:

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and Team work: Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

10. Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSO)
A graduate of the Computer Science and Engineering Program will demonstrate:

PSO1: Computer Science Specific Skills
The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

PSO2: Programming and Software Development Skills
The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

PSO3: Professional Skills
The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

Course Outcomes (CO)

After the completion of the course the student will be able to:

Course Outcome 1: Model and solve real world problems by applying knowledge across domains (Cognitive knowledge level: Apply).

Course Outcome 2: Develop products, processes or technologies for sustainable and socially relevant applications (Cognitive knowledge level: Apply).

Course Outcome 3: Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks (Cognitive knowledge level: Apply).

Course Outcome 4: Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms (Cognitive knowledge level: Apply).

Course Outcome 5: Identify technology/research gaps and propose innovative/creative solutions (Cognitive knowledge level: Analyze).

Course Outcome 6: Organize and communicate technical and scientific findings effectively in written and oral forms (Cognitive knowledge level: Apply).

Appendix C: CO-PO-PSO Mapping

COURSE OUTCOMES:

After completion of the course, the student will be able to:

SL.NO	DESCRIPTION	Bloom's Taxonomy Level
CO1	Model and solve real-world problems by applying knowledge across domains (Cognitive knowledge level:Apply).	Level3: Apply
CO2	Develop products, processes, or technologies for sustainable and socially relevant applications. (Cognitive knowledge level:Apply).	Level 3: Apply
CO3	Function effectively as an individual and as a leader in diverse teams and comprehend and execute designated tasks. (Cognitive knowledge level:Apply).	Level 3: Apply
CO4	Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms (Cognitive knowledge level:Apply).	Level 3: Apply
CO5	Identify technology/research gaps and propose innovative/creative solutions (Cognitive knowledge level:Analyze).	Level 4: Analyze
CO6	Organize and communicate technical and scientific findings effectively in written and oral forms (Cognitive knowledge level:Apply).	Level 3: Apply

CO-PO AND CO-PSO MAPPING

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	2	3							2	3	2	2
CO2	2	3	3	2	3	3	3	1	3			2		3	3
CO3			2		2				3	2	2			2	3
CO4			2	2	2			1	3	2	3	2		2	3
CO5	3	3	3	2	2							2	3		3
CO6					2			1	2	3	1	2			3

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

Mapping	Level	Justification
101003/CS822U.1-PO1	H	Understanding of fundamental engineering knowledge is essential for building the core functionality of the automated video translation system.
101003/CS822U.1-PO2	H	Problem analysis was crucial in identifying the language barrier and choosing appropriate technologies like NLP and TTS for solution design.
101003/CS822U.1-PO3	H	The system was designed considering the need for accuracy in synchronization and natural-sounding speech, addressing a real-world problem with societal relevance.
101003/CS822U.1-PO4	M	Used design of experiments and analysis for evaluating different speech-to-text and translation APIs.
101003/CS822U.1-PO5	H	Implemented and evaluated tools such as speech recognition, translation, and voice synthesis, demonstrating strong modern tool usage.
101003/CS822U.1-PO12	M	Required constant learning and adapting to new NLP models and TTS engines for better performance.
101003/CS822U.1-PSO1	H	Applied core Computer Science principles like NLP, deep learning, and signal processing to solve a domain-specific problem.
101003/CS822U.1-PSO2	M	Involved in programming and applying software development skills to build and integrate modules for transcription, translation, and synthesis.
101003/CS822U.1-PSO3	M	The project promotes professional growth through practical problem-solving and project management.
101003/CS822U.2-PO1	M	Applied systematic project development approach to improve functionality and accuracy of translation.
101003/CS822U.2-PO2	H	Researched and analyzed various APIs and frameworks for speech recognition and translation.

101003/CS822U.2-PO3	H	Designed and implemented a solution to a real-world language accessibility issue with multiple modules working in synchronization.
101003/CS822U.2-PO5	H	Integrated and evaluated performance of multiple modern tools (Google TTS, Whisper, etc.) within the application.
101003/CS822U.2-PO6	M	Considered user accessibility and societal impact, especially for non-English speakers.
101003/CS822U.2-PO7	M	Encourages content accessibility which supports cultural inclusion and sustainable media distribution.
101003/CS822U.2-PO8	L	Followed responsible practices during development and respected copyright implications while handling media.
101003/CS822U.2-PO9	H	Collaborated within a small team structure for system design and testing.
101003/CS822U.2-PO12	M	Involved continual research and adaptation to new tools and models during development, highlighting lifelong learning.
101003/CS822U.2-PSO2	H	Developed a real-time solution with multiple integrated algorithms, enhancing programming and software development skills.
101003/CS822U.2-PSO3	H	Engaged in testing and improving system usability and effectiveness, demonstrating professional competence.
101003/CS822U.3-PO9	H	Engaged in team discussions and group decisions during module integration and performance testing.
101003/CS822U.3-PO10	M	Prepared presentations and documentation to explain system working and highlight user benefits.
101003/CS822U.3-PO11	M	Managed system performance and integration timelines, showcasing basic project management skills.
101003/CS822U.3-PO12	M	Kept updated with changing AI APIs and cloud solutions, showing effort in lifelong learning.

101003/CS822U.3-PSO3	H	Gained practical exposure through testing and deploying models, developing professional and research-oriented skills.
101003/CS822U.4-PO5	M	Students used modern tools like speech recognition, translation APIs, and TTS systems effectively for implementation.
101003/CS822U.4-PO8	L	Followed responsible practices during development and respected copyright implications while handling media.
101003/CS822U.4-PO9	H	Collaborated within a small team structure for system design and testing.
101003/CS822U.4-PO10	M	Demonstrated communication skills by presenting project updates and explaining workflow to peers and evaluators.
101003/CS822U.4-PO11	M	Understood and applied basic project planning and scheduling principles.
101003/CS822U.4-PO12	M	Demonstrated consistent effort in improving the system based on feedback and adapting to new tech, indicating life-long learning.
101003/CS822U.4-PSO3	H	Worked collaboratively and showed initiative in building a usable, socially relevant product, demonstrating strong professional skills.
101003/CS822U.5-PO1	M	Applied domain knowledge in language processing and software development effectively.
101003/CS822U.5-PO2	H	Translated real-world needs into system features, designing a robust and scalable translation solution.
101003/CS822U.5-PO3	H	Proposed a clear problem definition and implemented a solution aligning with actual user needs.
101003/CS822U.5-PO4	M	Minor testing and analysis were done, though limited due to prototype-level implementation.
101003/CS822U.5-PO5	M	Applied speech-to-text, translation, and synthesis APIs appropriately with fair tool selection.

101003/CS822U.5-PO12	M	System improvement was mostly academic-focused; lifelong learning aspects were minimally involved.
101003/CS822U.5-PSO1	H	Addressed a grand challenge (language accessibility) using core CS principles in NLP.
101003/CS822U.6-PO5	M	Used open-source TTS and translation tools to build a working demo; technical awareness shown.
101003/CS822U.6-PO8	L	Basic understanding of copyright, privacy, and ethical issues was demonstrated.
101003/CS822U.6-PO9	H	Collaborated in peer groups for feature development, though work was mostly independent.
101003/CS822U.6-PO10	M	Created basic documentation and gave effective internal presentations.
101003/CS822U.6-PO11	M	Understood the basic scope of managing time and deliverables; handled tasks fairly well.
101003/CS822U.6-PO12	M	Showed initiative in learning about APIs, TTS models, and speech recognition through self-study.
101003/CS822U.6-PSO3	H	Demonstrated professionalism in delivering a working product that met social utility expectations.