

Project Proposal: Real Time English to Sinhala Dubbing System

Supervisor: Dr. Uthayasanker Thayasivam

Department of Computer Science and Engineering, University of Moratuwa

June 7, 2025

1 Introduction

The increasing demand for multilingual content access highlights the need for tools that make digital media more inclusive. One such need in Sri Lanka is the ability to access English video content in the Sinhala language, particularly for non-English speaking individuals. Our proposed system aims to bridge this gap by providing an automated solution that dubs English videos into Sinhala with synchronized speech.

1.1 Motivation

While subtitles are a popular option, they are not always accessible to individuals with visual impairments or reading difficulties. Moreover, real time consumption of content such as educational material or live broadcasts is hindered by subtitle limitations. This motivates us to build a system that enables seamless, voice-based content accessibility.

1.2 Project Objectives

- Develop an AI powered system that performs English to Sinhala video dubbing.
- Ensure intelligible, natural sounding output using advanced ASR, MT, and TTS technologies.
- Introduce support for multiple speakers and emotional tone in later phases.
- Enable future real time dubbing capability via streaming pipelines.

2 Related Work

Existing tools such as YouTube’s autocaptioning and subtitle translation rely mainly on text based solutions. Projects like Mozilla’s DeepSpeech, OpenAI Whisper, and Google Translate API offer components for transcription and translation, but do not provide a full dubbing pipeline. Commercial dubbing services are either manual or semi-automated and

lack support for underrepresented languages like Sinhala. Our work uniquely focuses on end to end automation of dubbing, with a target language that is typically low resource.

3 Proposed Approach

The system will be built in multiple phases:

3.1 Phase 1 - Core Dubbing Pipeline

- Audio is extracted from the uploaded video.
- Background music/noise is removed using source separation techniques.
- Speech is transcribed (Automatic Speech Recognition).
- The transcription is translated into Sinhala (Machine Translation).
- The translated text is synthesized (Text To Speech).
- The dubbed audio is merged back with the original video.

3.2 Phase 2 - Real Time Dubbing

- Audio is captured and processed in chunks for near real time performance.
- Streaming ASR, MT, and TTS services are connected via WebSockets.
- Latency and synchronization issues are addressed.

3.3 Phase 3 - Speaker Diarization, Voice Cloning, Emotion Detection

- Speaker diarization allows for multi-speaker support.
- Voice cloning can preserve the original speaker identity.
- Emotional tone can be detected and synthesized for expressiveness.

4 Expected Outcomes and Conclusion

This project will deliver a working prototype of an AI based English to Sinhala dubbing system. By the end of Phase 1, the core functionality of video dubbing will be achieved. Phases 2 and 3 will further improve usability, real time capability, and personalization. The final product will significantly improve content accessibility for Sinhala speakers.

5 Tools and Technologies

5.1 Speech Recognition (ASR)

- **OpenAI Whisper:** Multilingual ASR model with high accuracy, used for transcription.

5.2 Translation

- **Google Translate API:** For English to Sinhala translation.
- **Meta NLLB:** Alternative for low resource translation research.

5.3 Text-to-Speech (TTS)

- **OpenAI TTS:** High quality English speech synthesis. Explored for future Sinhala support.
- **Expressive TTS:** Tools such as Resemble.ai and Bark explored in Phase 3.

5.4 Video and Audio Processing

- **FFmpeg:** Used for audio extraction, background noise removal, and merging audio back with video.

5.5 Real Time Processing

- **WebSockets / WebRTC:** For real-time audio streaming in Phase 2.
- **Chunked pipeline processing:** For low latency inference.

5.6 Other Tools

- **React.js / Next.js:** Frontend development.
- **FastAPI / Node.js:** Backend APIs and model orchestration.
- **Docker:** Containerization.
- **GitHub:** Version control.
- **Draw.io / Mermaid.js:** Diagramming.
- **Overleaf:** Documentation.