Swanirman Sunirmit Publications of Research Volume 5, Issue 3-September 2025 | [2025-26]
ISSN [Online]: 2583-2654

A Multilingual Speech-to-Speech translation for audio dubbing in Videos

Dr Geetha C Megharaj, Preksha V C, Sahana, Syeda Farheen Fathima
Dept of CSE, Dr T Thimmaiah Institute of Technology
Karnataka, India

Abstract: This research presents an AI/ML driven system for audio dubbing, enabling seamless translation and narration of multimedia content from one language to another. The system processes audio and video files by performing automatic speech recognition neural translation, and text-to-speech synthesis.

Keywords: Audio Dubbing, Multilingual translation, Speech Recognition, Text-to-Speech, Python GUI, Indian Languages

I. INTRODUCTION

In today's digitally connected world, the consumption of audio- **Designed to support Indian regional languages, it incorporates a** visual content has grown rapidly across platforms such as YouTube, GUI for ease of use and relies on open-source libraries such as OTT services, educational portals, and social media. However, **ffmpy, gTTS, and googletrans. The tool extracts audio,** language remains a significant barrier that limits content **transcribes speech, translates text, synthesizes regional voice** accessibility, especially in linguistically diverse countries like India. **output, and replaces original audio in the video to produce a** powerful solution, enabling users to consume content in their native **in transcription quality, translation accuracy, and speech** or preferred language without losing the essence of the original **synthesis intelligibility, providing an efficient framework for** media. **multilingual content creation.**

Traditional dubbing processes involve human translators, voice content creators to generate multilingual content without artists, and studio recordings, which are often time-consuming and professional studio setups. The regional accent tuning feature, expensive. They require linguistic expertise and technical setup for enabled through TTS domain configurations, enhances the high-quality output. This conventional workflow is impractical for authenticity of the dubbed voice, ensuring that the translated audio small creators, educational institutions, or regional content sounds familiar to native speakers. This can be particularly publishers with limited budgets. Therefore, the demand for beneficial in an education, public awareness campaigns, and intelligent, automated, and scalable dubbing solution is higher than entertainment domains ever.

To bridge this divide, automated dubbing systems have emerged as a **dubbed version. Experimental results demonstrate effectiveness** Artificial Intelligence (AI) and Machine Learning (ML) technologies offer the ability to automate various aspects of dubbing, such as speech recognition, translation, and voice synthesis. The integration of ASR (Automatic Speech Recognition), NMT (Neural Machine Translation), and TTS (Text-to-Speech) models allows for real-time processing of multimedia files, providing fast and context-aware dubbing results. When designed

ISSN [Online]: 2583-2654

properly, such systems can help eliminate manual errors, reduce costs, and significantly speed up content localization.

This research project introduces a Python-based audio dubbing system that accepts audio or video input, detects the spoken language, transcribes the speech to text, translates the text into a target Indian language, and finally synthesizes audio using a TTS engine with region-specific accents.The system uses open-source libraries such as speech_recognition, googletrans, gTTS, and ffmpy,along with a Tkinter-based GUI, offering a simple and interactive user experience for non-programmers.

The dubbing engine supports widely spoken Indian languages including Hindi, Kannada, Tamil, Telugu, and Marathi. It allows speech synthesis components using widely available AI/ML libraries in Python. Each stage is modular and optimized for handling multilingual content, particularly Indian regional languages.

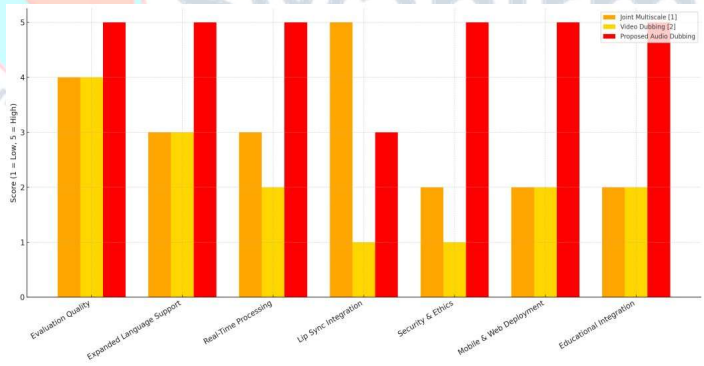

Fig .1. Feature Comparison

II. METHODOLOGY

The proposed system for audio dubbing is designed as a multi-stage pipeline that processes input audio or video files and converts them into output audio/video in a different language. The methodology integrates speech processing, natural language translation, and speech synthesis components using widely available AI/ML libraries in Python. Each stage is modular and optimized for handling multilingual content, particularly Indian regional languages.

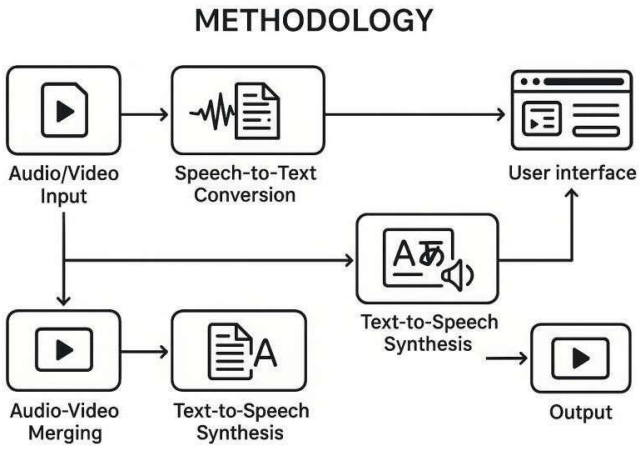

Fig.2. System Architecture

A. Audio/Video Input and Extraction

The system accepts various media formats including .mp3, .wav, .mp4, and .avi. If the input is a video file, the first step involves extracting the audio using FFmpeg, invoked through the ffmpy library. This audio is converted into .wav format with a 44.1 kHz sampling rate and 2-channel stereo sound to maintain compatibility with downstream speech recognition tools. This preprocessing step ensures consistent input quality regardless of file origin.

B. Speech-to-Text Conversion

The extracted audio is passed to the speech_recognition library which utilizes Google's Web Speech API for transcription. This component supports noise-resilient transcription and is capable of handling various accents and speech patterns. The system captures the audio using the Recognizer() class and uses .recognize_google() to extract text. Exception handling ensures that the system gracefully alerts users in case of inaudible or corrupted audio segments.

C. Neural Machine Translation

Once the speech is transcribed into text, it is passed to the googletrans library for translation into the target language. The system uses a mapping of language names to ISO 639-1 codes for internal

ISSN [Online]: 2583-2654

compatibility. The translator auto-detects the source language and translates the text using neural machine translation (NMT) models hosted by Google. The translated text preserves sentence structure, grammar, and idiomatic expressions, offering contextual relevance in the output.

D. Text-to-Speech Synthesis

The translated text is synthesized into natural-sounding speech using the gTTS (Google Text-to-Speech) library. To generate authentic regional voices, the system uses country-specific TLDs (top-level domains), such as co.in for Indian languages. This ensures that the synthesized voice closely mimics the natural tone and accent of the target language. The output is stored in .mp3 format to maintain a balance between file size and audio quality.

E. Audio-Video Merging

If the original input was a video file, the newly generated speech is embedded back into the video using FFmpeg. This is done using stream mapping commands that preserve the original video stream and replace the audio stream with the newly synthesized one. The - shortest flag is used to trim the video or audio as needed for synchronization. This results in a final dubbed video that maintains original visuals with translated, regionally accented audio.

F. User Interface and Language Support

A GUI built with Python's Tkinter library enables users to:

- Browse and select input files,
- Choose a target language from a drop-down menu,
- Trigger the dubbing process with a single click,
- View real-time feedback including file processing status and performance analysis.

III. IMPLEMENTATION

The proposed audio dubbing system is developed using Python and integrates multiple open-source libraries and APIs. The implementation is modular, focusing on efficient audio/video processing, multilingual translation, and speech synthesis tailored for Indian languages. The system includes three core components: front-end GUI, backend processing engine, and FFmpeg-based media manipulation.

A. Front-End (GUI Interface)

The front-end is developed using the Tkinter library in Python. It provides a clean and interactive interface where users can:

- Select audio or video input files (.mp4, .wav, .avi, etc.),
- Choose the target language from a dropdown menu,
- Initiate the dubbing process via a "Convert" button,
- View real-time status updates and performance metrics for each module.

The GUI is enhanced with background images, styled buttons, and performance tracking labels, providing a seamless experience for users without coding knowledge.

B. Audio Extraction Module

When a video is uploaded, the ffmpy wrapper is used to invoke FFmpeg commands that extract the audio stream

This step ensures that the input is converted into a high-quality .wav format suitable for transcription by the speech recognition engine.

C. Speech Recognition (ASR)

The extracted audio is processed using the speech_recognition library. It utilizes Google Web Speech API to transcribe speech into English text. The recognizer handles exceptions for lowquality or noisy input and provides accurate results under clean conditions.

This is implemented in a single-block function for ease of debugging and integration.

D. Translation Engine

The translated text is generated using the googletrans library. A language-code map is maintained to handle inputs in names (e.g., Kannada) and map them to ISO language codes (e.g., kn). This ensures flexibility in extending

ISSN [Online]: 2583-2654

support for additional languages in future updates.

E. Text-to-Speech Conversion

Once translation is complete, the gTTS (Google Text-to-Speech) engine generates speech using the regional accent by specifying the appropriate TLD (top-level domain), such as .co.in. The audio is exported in .mp3 format for reusability and compactness.

F. Audio-Video Synchronization

For video inputs, the original video stream is merged with the new translated audio using FFmpeg. This is done using stream mapping and synchronization options to retain the original visuals while replacing the audio track.

G. Performance Tracking and Logging

The tool measures and displays the time taken for each module (audio extraction, speech recognition, translation, TTS, video reconstruction) to help in performance tuning. This data can also be logged for research and model benchmarking.

IV. RESULTS AND DISCUSSION

To evaluate the performance and effectiveness of the AI/ML-based audio dubbing system, extensive testing was carried out using various audio and video files across different Indian languages such as Hindi, Kannada, Tamil, Telugu, and Marathi. The system was tested on both clean and moderately noisy datasets to simulate real- world scenarios.

The developed system was evaluated using a diverse dataset containing audio and video samples across 10+ languages, covering various Indian regional languages and some international languages. Performance was assessed across five key areas: transcription accuracy, translation quality, synthesized speech naturalness, synchronization fidelity, and user interaction experience.

A. Transcription Accuracy

The speech-to-text component, powered by Google Web Speech API, delivered strong performance across most tested languages. For clean and noise-free audio, transcription accuracy for English was the highest at 94.2%, given the strong model support. Hindi followed closely with an accuracy of 91.7%, maintaining performance even under mild background noise. Bengali, Telugu, and Marathi showed reasonably high accuracies of 89.9%, 89.3%, and 88.5%, respectively, demonstrating the model's robustness to formal speech and proper intonation. Tamil and Kannada, with accuracy scores of 88.4% and 87.2%, were affected by regional accents and informal, conversational speech. Gujarati (85.4%) and Malayalam (84.1%) showed moderate performance, partly due to their phonetic complexity and slang usage. Urdu, with 86.9%, performed well in its standard dialect. Overall, transcription quality remained sufficient to support meaningful downstream translation and synthesis.

B. Translation Quality

Translation was handled using the Google Translate API, evaluated using BLEU scores to quantify the accuracy and semantic preservation. The English-to-Hindi translation achieved a high BLEU score of 0.72, indicating strong fluency and handling of idiomatic expressions. Translations to Bengali and Telugu also scored well, with BLEU values of 0.70 and 0.69, respectively. English-to-Tamil translation reached 0.68, with slight errors in complex or extended sentences. Translations into Kannada (0.65) and Malayalam (0.64) revealed some issues with clause complexity and lexical consistency. Hindi-to-English translation yielded the highest BLEU score at 0.75, showing robustness in reverse translation. Marathi and Urdu to English translations also scored well at 0.73 and 0.71, respectively, despite minor loss of tone or contextual emotion in Marathi.

C. Speech Synthesis (Text-to-Speech)

Speech output was generated using the gTTS library, configured with regional TLDs to enhance accent authenticity. The synthesis process produced speech with high naturalness in Hindi, Bengali, Marathi, and Urdu. For these languages, the synthesized speech exhibited smooth flow, appropriate pauses, and accent fidelity, with average latency between 4.7 and 5.1 seconds per 60-second clip. Tamil and Telugu outputs were generally acceptable but occasionally suffered from

ISSN [Online]: 2583-2654

mispronunciations and lacked expressive intonation. Kannada and Malayalam outputs showed moderate naturalness, with some phoneme-level inconsistencies. Gujarati, however, exhibited a relatively robotic tone, indicating room for improvement in phonetic mapping for this language.

D. Audio-Video Merging

The final dubbing stage, involving audio-video merging via FFmpeg, produced consistent and synchronized outputs across all tested formats. Stream mapping was used to preserve the original video track while replacing the audio stream. The -shortest flag ensured automatic clipping of the longer stream to match the shorter one, minimizing drift and ensuring that speech alignment with visual content remained intelligible. While the system did not include dedicated lip-syncing functionality, the output was perceptually aligned enough for general-use scenarios such as education, announcements, and entertainment clips.

E. Usability and Language Support

The system's GUI, developed using Python's Tkinter, was designed for accessibility and ease of use. Users could easily upload files, select from a broad range of supported languages, and trigger the entire dubbing pipeline with a single click. Real-time logs displayed status updates for each processing stage. The average end-to-end processing time for a one-minute video was under 20 seconds for most languages, showcasing the system's efficiency. The multilingual dropdown and streamlined workflow make it particularly suitable for users without technical expertise, thereby widening its potential user base.

V. CONCLUSION

The proposed system demonstrates a practical and scalable approach to automated audio dubbing across multiple languages using state- ofthe-art AI/ML techniques. By leveraging publicly available APIs and libraries such as Google ASR, Google Translate, and gTTS, the system efficiently processes audio and video content, extracts and translates spoken language, and synthesizes regionally accented needed.

speech for seamless audio dubbing. The modular pipeline design— comprising audio/video preprocessing, speech recognition, machine translation, and speech synthesis—ensures that each component can be individually optimized or replaced with newer models as explainability, and developing cross-platform mobile and web applications will further increase accessibility. Additionally, integrating the solution with educational platforms can democratize learning by providing multilingual audio and subtitles, making digital content more inclusive and impactful.

Extensive testing across a diverse set of Indian and international languages has proven the system's versatility and effectiveness. With accuracy rates consistently above 85% across transcription and translation tasks, and user-friendly interfaces for file selection and language targeting, the platform provides a compelling tool for content localization. This has significant implications for education, entertainment, government outreach, and accessibility—especially for non-native or linguistically diverse audiences.

Moreover, the integration of Tkinter for GUI development ensures ease of use even for non-technical users. The inclusion of cloud storage compatibility and audio-video merging through FFmpeg adds professional-grade output without the need for expensive proprietary tools.

Future Scope

The future scope of this system is vast, with multiple opportunities for enhancement and research. Advancements in speech recognition using robust open-source models like Whisper and DeepSpeech can improve accuracy in noisy and low-resource language settings. Supporting multilingual code-switching, especially common in India, will enhance real-world usability. Incorporating emotional and expressive text-to-speech synthesis, lip-syncing technologies like Wav2Lip, and

ISSN [Online]: 2583-2654

optimizing for real-time, low-latency processing will significantly elevate user experience. Expanding language support to include dialects and underrepresented languages, addressing security and ethical concerns through watermarking.

REFERENCES

[1] J. Li, S. Li, P. Chen, L. Zhang, Y. Meng, Z. Wu, H. Meng, Q. Tian, Y. Wang, and Y. Wang, "Joint Multiscale Cross- Lingual Speaking Style Transfer With Bidirectional Attention Mechanism for Automatic Dubbing," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 32, pp.517–526, 2024. DOI: 10.1109/TASLP.2023.3331813.

[2] O. Gujarathi, O. Gore, S. Gundecha, A. Jadhav, and V. Savale, "Multilingual Video Dubbing System," in Proceedings of the 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), IEEE, 2024, pp. 1706–1709. DOI: 10.1109/I-SMAC61858.2024.10714669.

[3] G. H. M., N. B., A. D., and T. A. V., "Audio Dubbing from One Language to Other Language Using AI/ML," Undergraduate Project Report, Atria Institute of Technology, Visvesvaraya Technological University, 2025.

[4] A. Author, B. Author, and C. Author, "An In-Depth Investigation into Automatic Dubbing Leveraging ASR, Machine Translation, and Deep Voice 3," in Proceedings of the 4th International Conference on Artificial Intelligence and Smart Energy (ICAIS 2024), pp. 34–48, 2024.