PROJECT REPORT

Automated Video Dubbing: French to English

ACL Digital Thinkathon 2025

Theme: Shaping the Future: AI, Sustainability & Beyond

Submitted by:

Team Name: *NextBit Minds*

Team Members:

- Ayush Oza
- Gokul Bhalodiya
- Nirmit Vaidya
- Sneh Shah
- Soubhagya Sahoo

Table of contents

1. Executive Summary

In today's global digital ecosystem, language barriers significantly limit the accessibility and impact of multimedia content. The **Real-Time Video Translator** application addresses this challenge by enabling instant English translation of foreign-language videos through an interactive, user-friendly interface.

The core problem tackled in this project is the lack of fast, automated systems for **transcribing and translating video content in real-time**, especially for offline or personal use. Manual transcription and dubbing workflows are costly, slow, and inaccessible to most users.

1.1. Importance of Multilingual AI Accessibility

This project aligns with the growing need for **inclusive, AI-powered solutions** that break linguistic barriers. By leveraging multilingual AI models:

- Educational content becomes accessible to a broader audience.
- Localized communication is enabled for businesses and media.
- Users with hearing impairments or non-native language proficiency can benefit from synced subtitles.

Multilingual accessibility supports **sustainability** by reducing dependency on manual labor and reusability of content across regions and cultures.

1.2. Technologies Used

- **OpenAI Whisper:** For accurate, real-time speech-to-text and direct French-to-English translation.
- **FFmpeg:** For audio extraction from video in chunks.
- **Python + Tkinter:** For the cross-platform desktop GUI.
- **python-vlc:** For embedding VLC media player and synchronizing playback.
- **NumPy:** For audio signal processing.

1.3. Summary of Deliverables and Impact

- A real-time desktop application that:
  - Accepts local video files.
  - Transcribes and translates French audio to English on-the-fly.
  - Displays perfectly **synchronized subtitles** as the video plays.
- Modular code with support for threading and chunk-based processing.
- Delivered a usable prototype demonstrating how **AI can automate translation pipelines**, with immediate impact in **education, media localization**, and **personal content consumption**.

2. Introduction

2.1. Problem Background

The exponential growth of video content across platforms has created vast opportunities for learning, communication, and entertainment. However, **language remains a major barrier**—most videos are produced in a single language, limiting their reach and impact. Traditional solutions like manual dubbing and subtitle creation are time-consuming, costly, and require professional expertise.

This project aims to eliminate those limitations by providing an **automated system for translating and subtitling foreign-language videos in real time**. The solution is lightweight, offline-capable, and designed for individuals, educators, and content creators who need immediate and accessible translation.

2.2. Real-World Applications

This real-time video translator has practical use cases in a variety of domains:

- **Education:** Students can access lectures or tutorials in foreign languages with instant translation, making global knowledge more accessible.
- **Media Localization:** Journalists, filmmakers, and content creators can use the tool to preview or localize videos quickly without waiting for full dubbing cycles.
- **Travel and Cultural Exchange:** Tourists and language learners can use the tool to understand foreign informational or cultural media.
- **Accessibility:** Assists people who are hard of hearing or not fluent in the video's original language by offering translated subtitles in real time.

2.3. Relevance to Theme: AI, Sustainability, and Beyond

This project aligns directly with the hackathon theme:

- **AI:** Utilizes cutting-edge models like OpenAI Whisper to automate speech recognition and translation with high accuracy.
- **Sustainability:** Reduces the need for repeated human effort in content localization workflows—saving time, labor, and cost.
- **Beyond:** Opens doors for **real-time, offline, multilingual video interaction**, which goes beyond conventional subtitling and points toward next-gen AI-driven communication tools.

3. Objectives

The primary objective of this project is to develop a desktop-based application capable of **automating the translation and subtitle generation of French-language videos into English** in real-time. The application leverages state-of-the-art AI models to deliver fast, accurate, and synchronized subtitles as the video plays.

3.1. Key Objectives

3.1.1. Automate Speech Translation Workflow

Build an end-to-end pipeline that automatically:

- Extracts audio from a video file,
- Transcribes spoken French audio,
- Translates it into English, and
- Displays it as synchronized subtitles during playback.

3.1.2. Real-Time Performance

Ensure that the translation process operates in parallel with video playback by:

- Using chunked audio extraction (e.g., 10-second windows),
- Performing translation in a background thread,
- Starting video playback as soon as the first translated chunk is ready.

3.1.3. Maintain Natural Synchronization

Accurately map translated sentences to their corresponding audio timestamps to deliver natural, readable subtitles that stay in sync with the speaker.

3.1.4. Deliver a Lightweight, Usable Interface

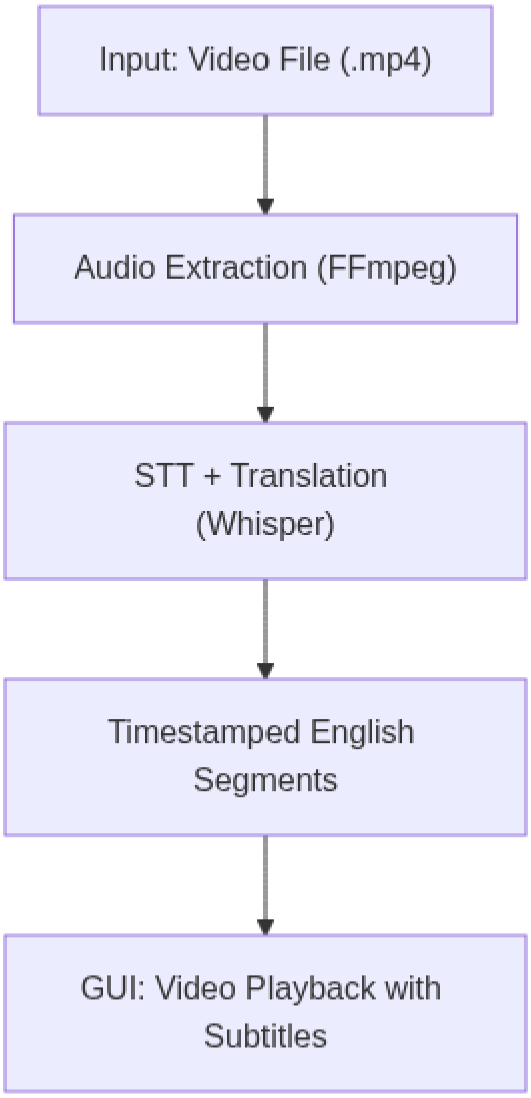
Provide a user-friendly desktop application with minimal setup requirements, allowing users to easily select a local video file and instantly see results without complex configurations.

3.1.5. Ensure Translation Quality and Context Preservation

Use robust AI models that preserve the intent, tone, and context of spoken language—even for conversational or idiomatic French—so that translations remain meaningful and fluent.

4. Project Architecture Overview

4.1. High-Level Pipeline Diagram


Input: Video File (.mp4)
Audio Extraction (FFmpeg)
STT + Translation (Whisper)
Timestamped English Segments
GUI: Video Playback with Subtitles

4.2. Component-Wise Breakdown

4.2.1. Audio Extraction

- **Purpose:** Extracts raw audio from the selected video in real time, one 10-second chunk at a time.
- **Tool Used:** ffmpeg (via ffmpeg-python)
- **How it works:**
  - Streams audio from the video using FFmpeg in s16le format.
  - Feeds each chunk directly to Whisper without writing intermediate files.
- **Relevant File:** translator.py

4.2.2. Speech-to-Text + Translation

- **Purpose:** Convert French audio into English text with timestamps in a single step.

- **Tool Used:** openai-whisper (task = "translate")
- **How it works:**
  - Loads the "base" Whisper model once.
  - For each chunk, performs transcription and translation.
  - Appends the result (start, end, text) to a shared global list.
- **Real-Time Behavior:** The first chunk is prioritized to start playback immediately.
- **Relevant File:** translator.py

4.2.3. Real-Time Subtitle Display

- **Purpose:** Display translated text segments in sync with video playback.
- **Tools Used:** tkinter (GUI) + python-vlc (video)
- **How it works:**
  - Monitors player.get_time() every 250 ms.
  - Checks current time against available segments.
  - Updates GUI label to reflect active translation.
- **Relevant File:** gui.py, video_player.py, utils.py

5. Speech Translation with Whisper

5.1. Purpose

The goal of this module is to convert spoken French audio from a video file directly into timestamped English text. This serves as the core of the automated video dubbing process, enabling accurate subtitle generation and cross-lingual accessibility.

5.2. Challenges

- **Noise and Background Interference:** Whisper must handle environmental noise, music, and overlapping sounds in the video.
- **Accent Variability:** The French speech may have regional accents that Whisper needs to interpret accurately.
- **Sentence Fragmentation Across Chunks:** Since audio is processed in 10-second segments, some sentences may be split between chunks, which could affect fluency and context.
- **Context Preservation:** Translating spoken language accurately requires understanding idioms, tone, and sentence-level context.

5.3. Model Used

- **Model:** OpenAI Whisper (base)
- **Mode:** task="translate"
- **Functionality:** Performs speech recognition and translation simultaneously (French audio → English text).
- **Rationale for Choosing:**
  - Open-source and easy to integrate.
  - Trained on multilingual data with strong performance in direct translation.
  - Handles timestamps automatically.
- **No intermediate French text is returned** — output is directly in English.

5.4. Preprocessing

- **Audio Format Conversion:**
  - Original audio extracted from the video using ffmpeg.
  - Converted to mono channel (ac=1), 16-bit signed PCM (s16le), 16kHz sample rate (ar=16000), which is Whisper's expected input.

- **Chunking Strategy:**
  - Audio is streamed in 10-second chunks.
  - Conversion is done in real-time using ffmpeg.run_async() with piped output.

5.5. Output Format

- The translated text is structured as a list of timestamped segments.
- Stored in segments_ref['list'], which is shared with the GUI for live subtitle rendering.
- Example:

None

```
{
  "start": 12.5,
  "end": 15.1,
  "text": "Welcome to the conference."
}
```

5.6. Advantages of Using Whisper's Integrated Translation

- Simplifies pipeline (no separate STT + translator).
- Improves performance due to tight sentence-to-sentence mapping.
- Reduces delay and improves real-time subtitle display.

6. Future Improvements

6.1. Lip-Sync Dubbing and Facial Animation

- Extend the system to support **text-to-speech (TTS)** synthesis and automatically overlay dubbed English audio onto the original video.
- Integrate with facial animation or **AI-driven avatar/lip-sync tools** (e.g., Wav2Lip, D-ID) to generate a fully dubbed version of the video that matches lip movements.
- This would eliminate the need for reading subtitles and enhance user immersion, especially in video content for education, storytelling, or entertainment.

6.2. Multilingual Support

- Expand the system to support **multiple source and target languages**, beyond French-to-English.
- Dynamically detect the input language and allow users to choose their preferred translation output.
- Leverage multilingual models like **Whisper large or NLLB (No Language Left Behind)** for broader language coverage.
- This would make the tool usable across global content, from Spanish lectures to Hindi documentaries.

6.3. Noise-Robust Transcription Models

- Integrate **denoising pre-processors** or train custom models to improve transcription performance in challenging environments (e.g., background music, overlapping voices).
- Experiment with **Whisper fine-tuning** or add speech enhancement modules (e.g., DeepFilterNet, RNNoise) for real-time cleanup of noisy inputs.
- This will make the tool more reliable for diverse media types and live recordings.

6.4. On-Device Real-Time Implementation

- Optimize the pipeline to run efficiently on **low-resource or edge devices** (e.g., Nvidia AGX orin dev kit).
- Reduce inference time by exploring **model quantization, distillation**, or replacing Whisper with faster lightweight STT models for portable use cases.
- Such an offline, privacy-preserving implementation would be ideal for classrooms, fieldwork, or regions with limited internet access.

7. Conclusion

7.1. Summary of What Was Achieved

The **Real-Time Video Translator** successfully demonstrates how advanced AI technologies can be integrated into a user-friendly desktop application to enable **live subtitle translation** for foreign-language video content.

The system:

- Accepts any local video file with French audio.
- Automatically extracts and transcribes speech.
- Translates it into English in real-time using **OpenAI Whisper**.
- Displays subtitles in perfect synchronization with the video via a **Tkinter + VLC GUI**.

This end-to-end pipeline was built with modular design, efficient resource usage, and a smooth user experience in mind. Users benefit from immediate playback, low latency translation, and intuitive controls.

7.2. Alignment with AI for Sustainability Goals

This project embodies the core goals of **AI for sustainability** by:

- **Reducing resource dependency** on manual transcription and translation services.
- **Promoting multilingual accessibility**, which allows content to be reused globally across language barriers.
- **Empowering inclusive communication**, especially for underserved communities and non-native speakers.
- Operating offline and efficiently, supporting **energy-conscious, edge-based** use cases in low-resource settings.

Through automation and scalability, the project contributes toward **sustainable digital content consumption and distribution**.

7.3. Final Remarks

This implementation proves the feasibility of **real-time, offline-compatible, AI-driven translation tools**. It showcases how speech and language models can be practically applied to solve accessibility problems today, with vast potential for future expansion into full dubbing, multilingual broadcasting, and adaptive learning.

The project provides a strong foundation for continued research and development in the fields of **AI-assisted communication, digital inclusion**, and **multimedia sustainability**.