AudioLens

Fall 2024 Cloud Computing Final Project

Group 6
Team members: Abid Azad, Jeff Acevedo
Professor: Maria Striki

***Abstract*— AudioLens is a mobile application that enhances accessibility and user engagement through real-time speech detection and translation from video files. Leveraging the Microsoft Azure Cognitive Speech Service SDK, the app processes speech from videos stored locally on a user's device, displays the detected speech above the playback interface and provides a translated English version below. This project was inspired by the Hawaii Toolkit, a previously popular tool for similar purposes, which is now deprecated in favor of Azure's ecosystem. Unlike Azure's Video Translator service, which requires a subscription, AudioLens utilizes the free-tier offerings of Azure's Speech Recognizer and Speech Translator services to provide cost-effective solutions for language accessibility. By focusing on open-source and affordable cloud-based technologies, AudioLens represents a practical application of cloud computing to address multilingual accessibility in personal media.**

Keywords*— *Software, IOS, Cloud Computing, Translation, Hawaii Toolkit, Azure

I. Recognize the need

The need for AudioLens arises from the growing prevalence of multimedia content in our daily lives and the challenges individuals face in accessing content in unfamiliar languages. In an increasingly globalized world, videos are locked across linguistic and cultural boundaries, making language barriers a significant obstacle. While automated translation services exist, they are often part of costly ecosystems or integrated into platforms that limit accessibility.

The Hawaii Toolkit, once a robust solution for real-time speech and translation tasks, provided users with accessible and efficient tools. However, its deprecation has left a gap in the market, necessitating an alternative that is both cost-effective and capable of leveraging modern advancements in cloud computing. Microsoft Azure offers a suite of speech services that can bridge this gap. However, the Video Translator service is a premium tool, limiting its adoption among users or developers operating within financial constraints.

AudioLens addresses this need by utilizing Azure's free-tier Speech Recognizer and Speech Translator services to process video files stored locally on mobile devices. This approach provides access to advanced speech and translation technologies and aligns with the principles of cost-effectiveness and inclusivity. The app empowers users to understand and interact with video content in different languages, enhancing accessibility and breaking down language barriers for a global audience.

II. Background

The AudioLens project employs advanced cloud computing services to deliver real-time speech recognition and translation for videos locally stored on mobile devices. By leveraging Microsoft Azure's Cognitive Speech Services, AudioLens addresses a clear need for multilingual accessibility while taking advantage of modern cloud computing advancements as follows.

A) The Hawaii Toolkit: A Legacy Solution

The Hawaii Toolkit was a groundbreaking project developed by Microsoft to empower mobile and cloud-based applications. It provides APIs for real-time services such as speech recognition, translation, and image processing. These capabilities enabled developers to build applications that bridged communication gaps across languages, supported real-time interactions, and enriched user experiences with cloud-connected tools.

Key features of the Hawaii Toolkit included:

- Speech Recognition and Translation: The toolkit provided APIs that could convert spoken language into text and translate it into a target language. This laid the groundwork for many accessibility-focused applications.

- Real-Time Connectivity: The Hawaii Toolkit relied on cloud computing to process data in real-time, ensuring that applications using the service could deliver immediate results.

- Ease of Use for Mobile Developers: Designed with simplicity, the toolkit enabled seamless integration with mobile platforms, especially iOS and Android.

However, with Microsoft's pivot to the Azure ecosystem, the Hawaii Toolkit was discontinued in favor of a more comprehensive cloud-based infrastructure. While this transition brought more advanced tools like the Azure Video Translator, it also created challenges for developers seeking cost-effective solutions, as many Azure services are locked behind paywalls. AudioLens draws inspiration from the Hawaii Toolkit but adapts to modern needs by utilizing Azure's free-tier services, making the project viable for users without significant financial resources.

B) Leveraging Cloud Computing in AudioLens

Cloud computing plays a pivotal role in AudioLens by offloading audio processing and translation tasks to Azure's global infrastructure. Here is how the cloud enhances the functionality of the two primary services:

- Centralized Processing: AudioLens sends raw audio data from video playback to Azure's servers. The cloud infrastructure processes this data using pre-trained speech recognition and translation models. This eliminates the need for local storage of models or computationally intensive processing on the mobile device.

- Scalable Machine Learning Models: Azure's Speech Recognizer and Translator services use pre-trained deep neural networks hosted in the cloud. These models are constantly updated with new data to improve accuracy. AudioLens automatically benefits from these updates without requiring any intervention from the developer or user.

- Global Low-Latency Network: By leveraging Azure's globally distributed data centers, AudioLens minimizes latency. Audio is processed in near real-time, with the detected and translated text returned to the app within milliseconds.

- Cost Efficiency: Instead of relying on premium paid APIs like Azure's Video Translator, AudioLens uses free-tier speech recognition and translation services. These services offer ample capabilities for personal use without additional financial burden.

C) Microsoft Speech Recognizer: Cloud Computing and Configurations

The Microsoft Speech Recognizer API converts spoken audio into text and is powered by cloud computing and SPX configurations.

How It Works:

- Audio Data Transmission: AudioLens captures the audio stream and sends it to Azure's servers when a video is played. This step uses SPX Audio Configuration, which defines the input source (e.g., a microphone or a local audio file). For AudioLens, this configuration ensures that the audio extracted from the video file is streamed seamlessly to Azure.

- SPX Auto Detect Configuration:
    - The Speech Recognizer uses SPX Auto Detect Source Language Configuration, allowing it to identify the audio's language dynamically. This feature is critical

    for handling multilingual videos without requiring manual language selection.
    - During initialization, the app specifies a set of supported languages for the recognizer to detect. The service then evaluates the incoming audio to determine the spoken language before transcribing it.
    - For example, if a video contains French and English, the auto-detect configuration ensures the recognizer can switch between the two seamlessly without additional user input.

- Cloud-Based Recognition: Once the audio reaches Azure, it is processed using cloud-hosted acoustic and language models. These models analyze phonemes, syntax, and context to generate a high-accuracy transcription.

- SPX Speech Configuration: The Speech Recognizer is initialized using SPX Speech Configuration, which specifies the Azure subscription key and service region. This configuration ensures secure authentication and directs the audio data to the appropriate Azure endpoint.

D) Microsoft Speech Translator: Cloud Computing and Configurations

The Microsoft Speech Translator API takes the transcribed text from the Speech Recognizer and converts it into the desired target language using Azure's cloud services.

How It Works:

- Input from Speech Recognizer: The Speech Translator receives text output from the Speech Recognizer. The use of SPX configurations ensures seamless data flow between the two services.

- SPX Speech Configuration:
    - Like the Speech Recognizer, the Speech Translator is initialized using SPX Speech Configuration to authenticate the Azure service and define the cloud endpoint.
    - In this configuration, the app specifies the source language (auto-detected by the recognizer) and the target language (e.g., English).

- Translation in the Cloud:
    - The text is sent to Azure's neural machine translation models hosted in the cloud. These models analyze grammar, syntax, and idiomatic expressions to produce contextually accurate translations.
    - Azure's NMT models are continually trained and optimized using massive multilingual datasets. AudioLens benefits from these improvements without requiring updates to the app itself.

- SPX Audio Configuration: The Speech Translator does not directly handle audio data but relies on the transcription provided by the Speech Recognizer. The audio configuration ensures the raw audio stream is effectively captured and passed along.

Real-Time Feedback:

Once the translation is complete, the service sends the translated text back to the app chunk by chunk, just like the transcription. This allows the app to display translations below the video playback in real time, keeping pace with the ongoing audio.

E) How AudioLens Streams and Processes Audio with Cloud Computing

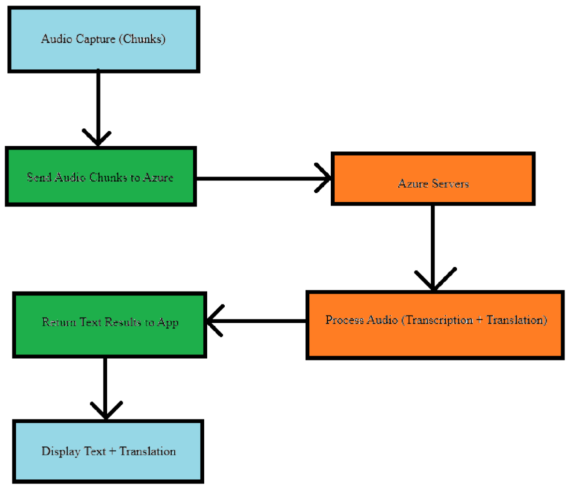
The real-time speech recognition and translation in AudioLens rely on breaking the audio stream into manageable chunks, processing these chunks incrementally, and leveraging Azure's cloud infrastructure to return results dynamically. This approach ensures low latency and real-time feedback for the user. Below is a detailed explanation of the process.

- Audio Capture and Chunking: Audio is captured and split into chunks (200–500 milliseconds). These chunks are streamed to Azure servers incrementally using a WebSocket or HTTPS connection.

- Cloud-Based Processing: Each chunk is processed independently in Azure's servers for phoneme mapping, language detection, transcription, and translation.

- Returning Results: Transcription and translation results are returned to the app chunk-by-chunk, ensuring real-time updates synchronized with the video playback.

- Dynamic Adaptation: With SPX Auto Detect Source Language Configuration, the app adapts seamlessly to multilingual inputs without user intervention.


Audio Capture (Chunks)
Send Audio Chunks to Azure
Azure Servers
Process Audio (Transcription + Translation)
Return Text Results to App
Display Text + Translation

This chunk-based, cloud-driven process underpins the efficiency and accuracy of AudioLens, making it an ideal solution for real-time speech recognition and translation.

III. Design

The design of AudioLens combines efficient video playback, real-time cloud-based speech recognition and translation, and a user-friendly interface to ensure seamless accessibility for multilingual video content. This section describes the app's structure, detailed functionality, and UI design to understand how it works clearly.

A) Architectural Overview

AudioLens integrates three primary modules:

1) Video Management: Handles video selection, playback, and audio extraction.

2) Speech Processing: Microsoft Azure Cognitive Services is used for language detection, transcription, and translation.

3) User Interface (UI): Displays the video along with detected and translated text in real-time, accompanied by intuitive playback controls.

Cloud computing ensures the app remains lightweight by offloading complex tasks to Azure's globally distributed servers, enabling real-time results even on mobile devices.

B) Core Functionalities

Video Management

AudioLens allows users to select a video from their device's library and handles the following tasks:

Video Selection: Users choose a video using the "Select Video" button. The app opens the device's photo library and filters for video files.

```
On "Select Video" button click:
  Open photo library
  Allow users to pick a video
  If video selected:
    Extract audio from video
    Start playback
```

Video Playback: The app uses AVPlayer to play the selected video and displays it in a resizable video frame on the UI.

```
Load video into video player
Display video on screen
Start playback
```

Audio Extraction: The app extracts audio from the video for processing. If the video lacks an audio track, the app displays an error message

```
Extract the audio track from video

Save audio as a temporary file

If no audio found:

    Display error message
```

Speech Processing

Once audio is extracted, the app processes it using Azure’s Speech Recognizer and Speech Translator APIs.

- Language Detection: The app uses SPXAutoDetectSourceLanguageConfiguration to identify spoken language from audio files automatically. This eliminates the need for users to specify the language.

```
Send extracted audio to Azure

Use auto-detection for supported languages

Return detected language
```

- Real-Time Speech Recognition and Translation: Detected speech is sent to Azure’s SPXTranslationRecognizer, which translates the text into English in real-time. The app then streams audio to Azure on a chunk-by-chunk basis, using Azure’s SPXPushAudioInputStream, enabling immediate feedback.

```
For each audio chunk:

    Send chunk to Azure Speech Recognizer

    Receive transcribed text

    Display text above video



For each transcribed text:

    Send text to Azure Translator

    Receive translated text

    Display translation below video



While video is playing:

    Extract audio chunks

    Send chunks to Azure for processing

    Receive and update transcription/translation
```

User Interface

The UI is designed to provide clear, accessible feedback and control for users:

- Video Display: A prominent video frame in the center of the screen displays the selected video. Below the video, playback controls (Play/Pause, Rewind) allow users to navigate.

- Detected Speech: In real-time, a text box above the video displays the detected speech in the original language. The detected text is dynamically updated as the video plays.

- Translated Speech: A text box below the video shows the translation into English, updated alongside the detected speech.

- UI Elements: The app uses intuitive labels, buttons, and a dark theme to enhance readability and usability:
    - "Select Video" Button: Positioned in the top-right corner, allowing users to upload a video.

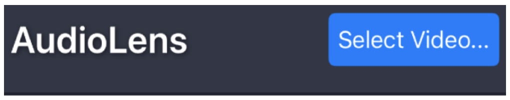
    - "Detected Text" and "Translated Text" Sections: Clearly labeled areas provide real-time feedback.

    - Loading Overlay: Displays a processing indicator when the app analyzes audio or streaming to Azure.

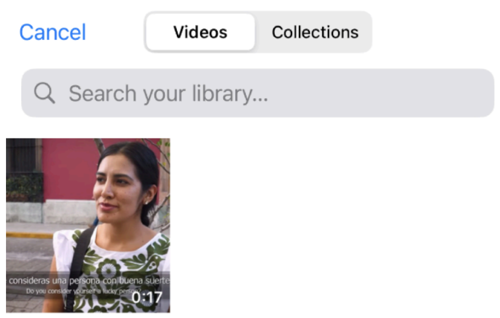
```
UI Layout:
  Header:
    - App title
    - "Select Video" button
  Middle Section:
    - Video player
    - Playback controls (Play/Pause, Rewind)
  Detected Text Section:
    - Real-time transcription above the video
  Translated Text Section:
    - Real-time translation below the video

On audio processing:
  If transcription available:
    Update "Detected Text" box
  If translation available:
    Update "Translated Text" box
```

Screenshots of UI

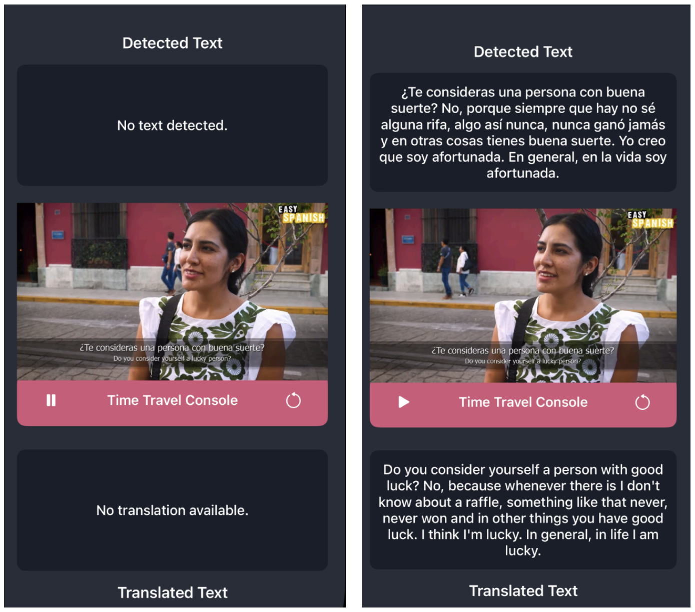
Select Video Button:



Video Selection Screen:



Home Page w/ Video before and after Translation:



Detected Text

No text detected.

No translation available.

Translated Text

Detected Text

¿Te consideras una persona con buena suerte? No, porque siempre que hay no sé alguna rifa, algo así nunca, nunca ganó jamás y en otras cosas tienes buena suerte. Yo creo que soy afortunada. En general, en la vida soy afortunada.

Do you consider yourself a person with good luck? No, because whenever there is I don't know about a raffle, something like that never, never won and in other things you have good luck. I think I'm lucky. In general, in life I am lucky.

Translated Text

IV. Conclusion

AudioLens manages to capture the complexity and efficiency of a real-time audio translator by utilizing Azure’s free-tier cloud based services. By using the discontinued Hawaii Toolkit as a foundation for our design and then the Azure services available to us, such as Speech Recognizer and Translator, to achieve near accurate results as shown in the right-most screenshot of the UI above.

However, a shortcoming we did come across was the fact that it seemed very inefficient for both the system and us, the developers, to use two different services when one is very similar to the other. The speech recognizer and the speech translator are very similar to each other in terms of set up, however the translator service does not come with an auto-detect configuration unlike the recognizer. Using these two services together takes up more cloud resources, increased latency and processing time, and the system provides more effort than it needs to with repeated and/or redundant steps. A good way to combat this would be to implement the auto detect feature onto the translator service. If that won’t work, it might be best to send only a small little chunk of audio, like we did in the translator, to the recognizer, which will minimize the duplicated effort and save up resources.