MULTI-LANGUAGE LIPSYNC

A thesis submitted in partial fulfillment of the requirements for the award of the degree of

B.Tech in

Computer Science and Engineering

By

Khushi Goel (106121064)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY TIRUCHIRAPPALLI - 620015

2023

Abstract

According to recent surveys, content consumption is currently hovering around 3.5, which is believed to have grown significantly in recent years. The viewers streaming the content are counted in billions. On average it is calculated that each user spends 17 hours consuming the content on the internet every week.

To cater a larger number of audiences many companies and studios are dubbing entertainment purpose movies and educational videos into other languages but along with that there comes in many other issues out of which one is the main hinderance I.e., the lips doesn't get synced to the video which sometimes breaks immersion of the consumer.

There has been research to automate this process to minimize manual work. Successful paperwork is known to achieve this but with the constraint of having only one language.

To be approximate there are 7000 languages spoken in the world so manually it's not practical and possible to train everyone and hence this initiative has been taken to make speaker and language both independent.

Through this project we intend to make the model more conveniently understandable so that the model can perform lip sync in a better way.

Problem Statement

Currently, the SSML (Speech Synthesis Markup Language) approach is used for lip synchronization which is bound to work only when there are known and popular languages. The existing model faces the problem of implementing the effectiveness of SSML in ensuring accurate pronunciation and naturalness in uncommon languages and dialects.

To optimize it, we intend to add the feature of timings at the given current time stamp so that there is a proper lip synchronization component to the model.

Input/Output

Input:

Single speaker Audio/Video: The primary input is an audio/video i.e. single speaker speaking

Video: In the case of the video dataset, we use this advantage to enhance the lip movements and synchronization with the present speaker, we have pre-downloaded videos.

Output:

Lip-Synced Video: The key outcome is a video in which the lips of the identified current timestamp face move in time with the audio track.

Dataset:

Step 1: Upload Video

Step2: Download SRT(SubRip Subtitle file) files with different languages you intend to dub in

Step 3: Organize the Dataset (SRT files)

Step 4: Use for Testing

Objectives

We aim to achieve the following objectives through this project:

1. To research the existing solutions and approaches regarding the problem in the market.

2. To conduct speaker identification and reduce the manual work of speaker identification.

3. In addition to the above point, we aim at smooth lip synchronization for uncommon languages

Literature Survey:

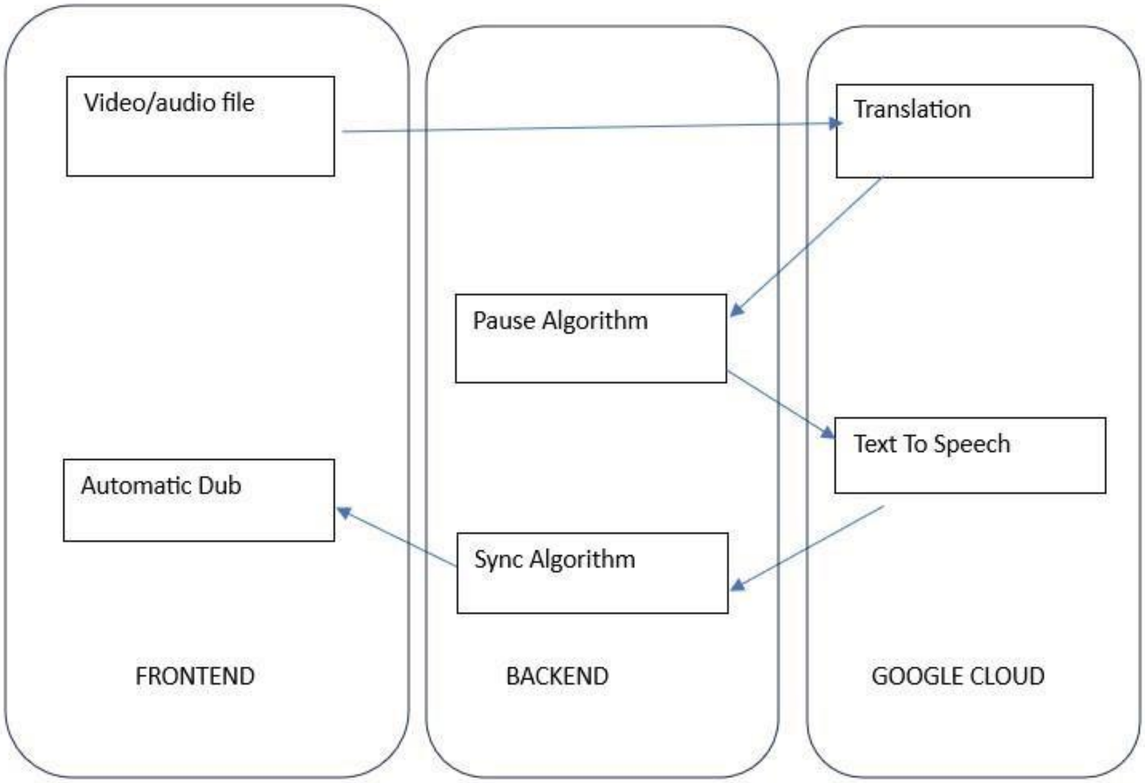
Paper	Publication	Concept	Conclusion/Limitation
A Survey of Research on Lipreading Technology	IEEE ACCESS November 9, 2020.	The survey covers traditional and deep learning lipreading methods, databases, and recognition rates, alongside challenges and future research directions.	The survey distinguishes itself from previous reviews by providing a comprehensive examination of both the frontend and back-end network structures in deep learningbased lipreading.
Towards Automatic Faceto-Face Translation	Multimedia, ACM, 2019	It gauges how much lip synchronization there is between the frames.	The lip-sync accuracy of lipGAN about only 50 percent although it tries to aim at achieving the speaker independence.
A Lip Sync Expert Is All You Need for Speech to Lip Generation in the Wild.	Multimedia, ACM, 2020	It uses a pre trained lip sync expert.	The current timestamp speaker and dubbing sync is independent known to set new metrics for real world. In case of change in lighting, it cannot handle sudden change in frame

VideoReTalking: Audio-based Lip Synchronization for Talking Head Video Editing in the Wild	SIGGRAPH Asia 2022	Prior to generating the lips, neutralize the facial expressions and then utilize the changed frames as posture guides. Finally, utilize a Style GANbased restoration and an identityaware augmentation network to achieve photo-realistic results.	Produce top-notch outcomes with emotional talking-head generation for the lower half of the film. Modify the identity of the character somewhat from the original video.
Visual dubbing pipeline with localized lip-sync and two-pass identity transfer	Computers and Graphics, Volume 110, 2023.	It includes three main steps for proposing the algorithm: pose alignment, identity transfer, and video reassembly	It is specifically designed to keep the original video intact. For identity transfer, it requires another video for the destination /final output

Research Gaps and Challenges

Any of the current existing models have not been able to achieve and solve the general issue of timings, specifically how we match the length of various words in languages after dubbing. When using models like easydubz, the resultant /output video helps us realize that there is some lag between the dialects and the video playing. The viewers might find a break in immersion because of the lack of lipsynchronization as a result.

BLOCK DIAGRAM -



METHODOLOGY

1. **User Uploads Video with Captions:**
   - Users upload a video file along with caption text in a source language to the website.
2. **Server Communication with Google Cloud API:**
   - The server communicates with the Google Cloud API to perform various tasks, including:
   - Audio transcription (speech-to-text) to convert spoken words in the video to text.
   - Language detection to identify the source language of the captions.

- Caption parsing and timing calculations to segment the text into translation units with appropriate pauses.

3. **Translation and SSML Generation:**
   - The caption text is translated from the source language to the target language(s) using the Google Translate API.
   - The translated text is used to generate Speech Synthesis Markup Language (SSML) that includes timing information for speech pauses. This SSML is important for ensuring the generated audio is synchronized with the video.
4. **Google Text-to-Speech API:**
   - The SSML content is sent to the Google Text-to-Speech API to convert the translated text into audio in the target language(s). This API generates audio files that include prosody information for natural-sounding speech.
5. **Adjusting Audio Sampling Rate:**
   - The audio files generated by the Text-to-Speech API may have different sampling rates than the original video. You need to adjust the sampling rate to match the video's length. This may involve resampling the audio.
6. **Sending Adjusted Audio to the Browser:**
   - The adjusted audio files are sent back to the user's browser for playback. The browser plays the video and syncs the audio with the captions, providing a multi-language lip sync experience.

ALGORITHMS:

COLOUR CODES: ALGORITHM USED; ALGORITHM IMPLEMENTED BY US

1. Flask Routing Algorithm:

The Flask web framework routes HTTP requests to specific functions based on URL endpoints, allowing the code to handle different actions based on the route.

2. Google Cloud Translation API:

The code uses the Google Cloud Translation API to translate text from one language to another. While not explicitly shown in the code, the API employs various algorithms for translation.

3. Google Cloud Text-to-Speech API:

The code uses the Google Cloud Text-to-Speech API to convert text into speech. The API likely uses complex algorithms for generating speech from text.

4. Synchronization Algorithm:

The sync function adjusts the playback speed of the audio to synchronize it with the subtitle duration. While not detailed in the code snippet, this synchronization involves altering the audio speed and timing.

5. Text Breakdown Algorithm:

The break_apart function divides long text into smaller segments (subtitles) to ensure that each segment does not exceed the Text-to-Speech API's character limit. The algorithm finds suitable breakpoints in the text.

6. File Handling Algorithms:

The code handles file uploads and serves files for download using Flask's send_from_directory function. These operations involve file I/O and directory management.

7. Speech Generation Algorithm:

The core of speech generation is handled by the Google Cloud Text-to-Speech API, which employs algorithms for converting text into speech. The code coordinates the usage of this API.

8. Main Functionality Algorithm:

The if __name__ == '__main__': block is not an algorithm itself but rather a section of code that tests the core functionality of the application. It demonstrates how to use the SRTtoAPI function to convert subtitles to speech.

9.Language Selection Algorithm:

The code selects the appropriate language and voice for translation and speech generation based on the user's input and predefined language dictionaries.

10.Translation Algorithm:

The code uses the Google Cloud Translation API for translating text. The algorithm for translation is abstracted behind the API.

11.SRT Parsing Algorithm:

The Caption_Conversion.SRT_to_API function likely includes an algorithm for parsing SRT subtitle files and extracting text.

12.SRT Formatting Algorithm:

The Caption_Conversion.SRT_to_API function may also include algorithms for formatting the extracted text for further processing.

OUR MODIFICATIONS.

1.Neural Networks

A neural network is a kind of machine learning method that mimics the way the human brain works, using computer science and statistics to find patterns in data for AI problems according to the requirements we can apply different optimizations. In the human brain, a neural network is a group of biological neurons that send signals to each other using electricity and chemicals. A neural network is made of layers of nodes or artificial neurons that handle and pass on information. Each node has a weight and a threshold that decide how much it affects the output. The output of one node is the input of another node in the next layer. The network learns from data by changing the weights and thresholds based on feedback. Neural networks can do tasks like recognizing speech, analyzing images, controlling systems, and more.

DEEP LEARNING APPROACH:

Features which deep learning algorithms aims to work on :

1. **Language-Specific Timing Models:** Deep learning models can be trained to capture the unique timing patterns and problem of uncommon languages.

2. **Phoneme-Level Alignment:** Deep learning models can align phonemes in the source and translated speech.This is essential for maintaining natural lip sync.

3. **Prosody Prediction:** Deep learning models can predict prosodic features, such as pitch, stress, and rhythm, which are crucial for determining speech timing.

4. **Custom SSML Markup Generation:** Deep learning models can generate custom Speech Synthesis Markup Language (SSML) with precise timing

instructions, adapting speech rate, pause durations, and emphasis as needed for uncommon languages.

We primarily work with involving text-to-speech (TTS) conversion and timing.

METHODOLOGY OF MODIFICATION

1. **Data Preparation:**
    - Gather a multilingual dataset containing videos with speakers from English, Bangla, and Hindi. Ensure that the dataset includes corresponding SRT files with accurate timing information for the spoken words.
2. **SRT Parsing: pause algorithm**
    - Develop a parser to extract the text and timing information from the SRT files. This information will be crucial for aligning the lip movements with the spoken words.
3. **Multimodal Model Design:**
    - Design a multimodal deep learning model that takes both the lip images and the corresponding text as input. This could involve a combination of (CNNs) for processing lip images and (RNNs) or transformers for processing the textual information.
4. **Transfer Learning and Fine-Tuning:**
    - Consider using transfer learning techniques. You might pre-train the model on a large dataset containing lip synchronization information from one language and then fine-tune it on your multilingual dataset.
5. **Web Application Integration:**

- Develop a web application where users can upload SRT files and source language videos. Integrate the deep learning model into the backend of the application to perform lip synchronization.

6. **Inference and Results:**

    - Implement the inference process in your web application. Given an uploaded SRT file and video, use the trained model to synchronize the lip movements with the spoken words. Provide the results to the user.

EVALUATION METRICS

1. **Timing Precision in Uncommon Languages:**

    - Assess the precision of timing adjustments in uncommon languages. Calculate the average timing deviation (early or delayed) for each segment or word in these languages.

2. **Word Error Rate:**

Components of WER Calculation:

Insertions (I):

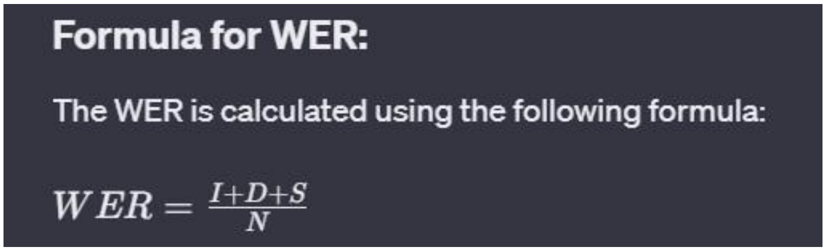
The number of words in the hypothesis that are not present in the reference.

Deletions (D):

The number of words in the reference that are not present in the hypothesis.

Substitutions (S):

The number of words in the hypothesis that are different from the corresponding words in the reference.



Formula for WER:

The WER is calculated using the following formula:

$$WER = \frac{I+D+S}{N}$$

Understanding WER Interpretation:

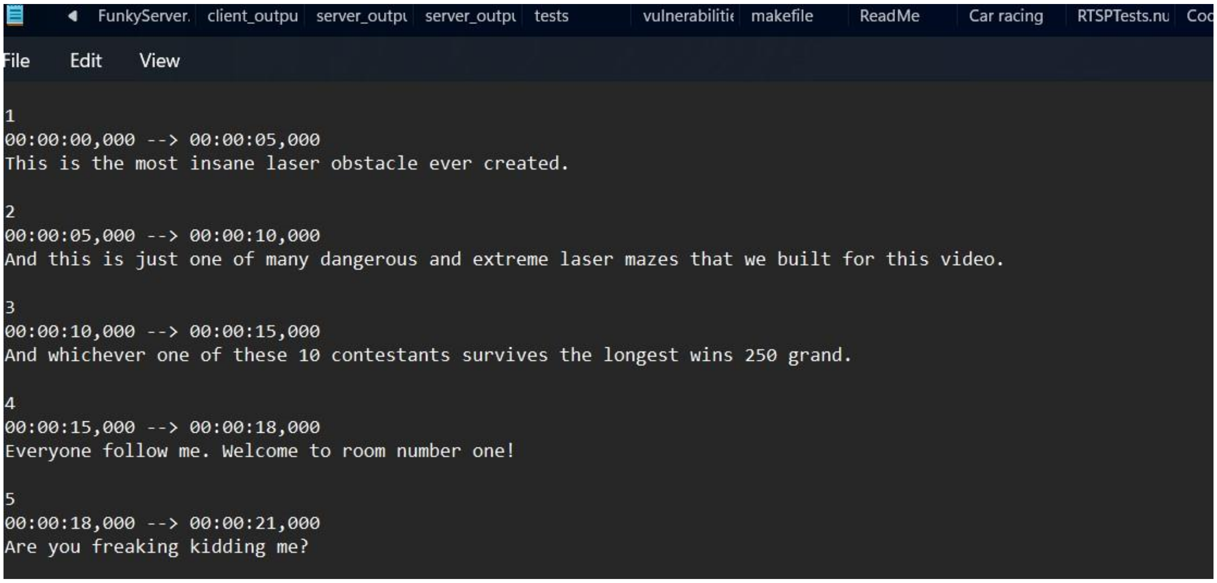
WER = 0%: Ideal case, where the hypothesis exactly matches the reference (perfect transcription).

WER > 0%: Indicates errors in the transcription. The higher the WER, the more errors there are relative to the length of the reference.

Results and Test Cases

1. **Test Case:** Upload an English video with a corresponding SRT file containing accurate transcriptions and timestamps.

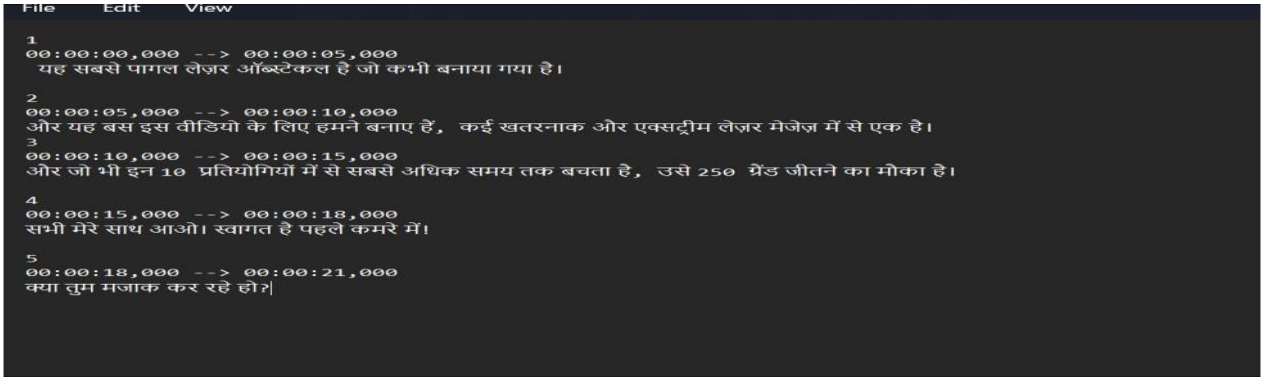
Expected Result: The system accurately synchronizes lip movements with spoken words in English.



```
1
00:00:00,000 --> 00:00:05,000
This is the most insane laser obstacle ever created.

2
00:00:05,000 --> 00:00:10,000
And this is just one of many dangerous and extreme laser mazes that we built for this video.

3
00:00:10,000 --> 00:00:15,000
And whichever one of these 10 contestants survives the longest wins 250 grand.

4
00:00:15,000 --> 00:00:18,000
Everyone follow me. Welcome to room number one!

5
00:00:18,000 --> 00:00:21,000
Are you freaking kidding me?
```

2. **Test Case:** Upload a video containing segments in English, Bangla, and Hindi with corresponding SRT files.

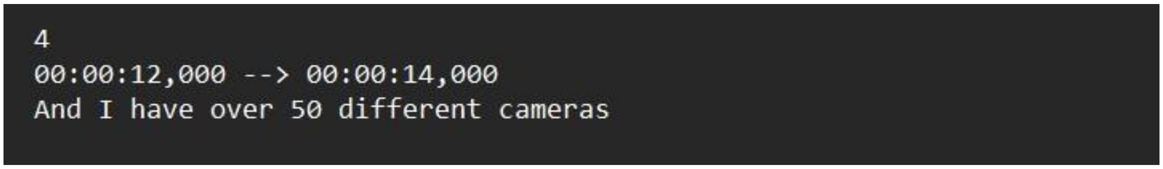
Expected Result: The system accurately synchronizes lip movements for all three languages within the same video.



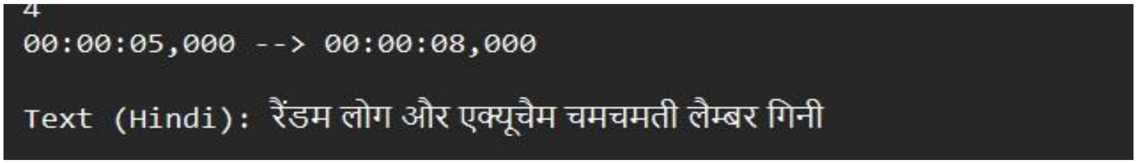
```
1
00:00:00,000 --> 00:00:05,000
 यह सबसे पागल लेज़र ऑब्स्टेकल है जो कभी बनाया गया है।

2
00:00:05,000 --> 00:00:10,000
और यह बस इस वीडियो के लिए हमने बनाए हैं,  कई खतरनाक और एक्सट्रीम लेज़र मेजेज़ में से एक है।
3
00:00:10,000 --> 00:00:15,000
और जो भी इन 10  प्रतियोगियों में से सबसे अधिक समय तक बचता है,  उसे 250  ग्रेंड जीतने का मौका है।

4
00:00:15,000 --> 00:00:18,000
सभी मेरे साथ आओ। स्वागत है पहले कमरे में!

5
00:00:18,000 --> 00:00:21,000
क्या तुम मजाक कर रहे हो?
```

3. **Test Case**: Upload a video with captions in Bangla, Hindi, or English and assess the timing accuracy of the generated audio.

Expected Result: The system should effectively handle these languages and provide precise timing adjustments



```
4
00:00:12,000 --> 00:00:14,000
And I have over 50 different cameras
```



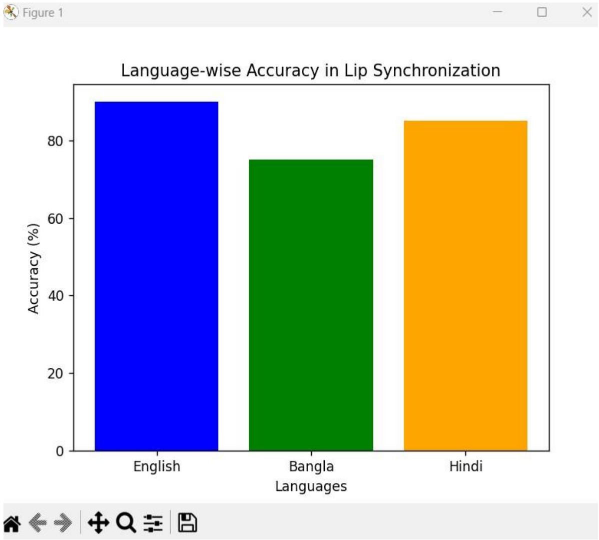
```
4
00:00:05,000 --> 00:00:08,000

Text (Hindi): रैंडम लोग और एक्यूचैम चमचमती लैम्बर गिनी
```

ANALYSIS

1.Language Wise accuracy :

X-axis: Languages (English, Bangla, Hindi)

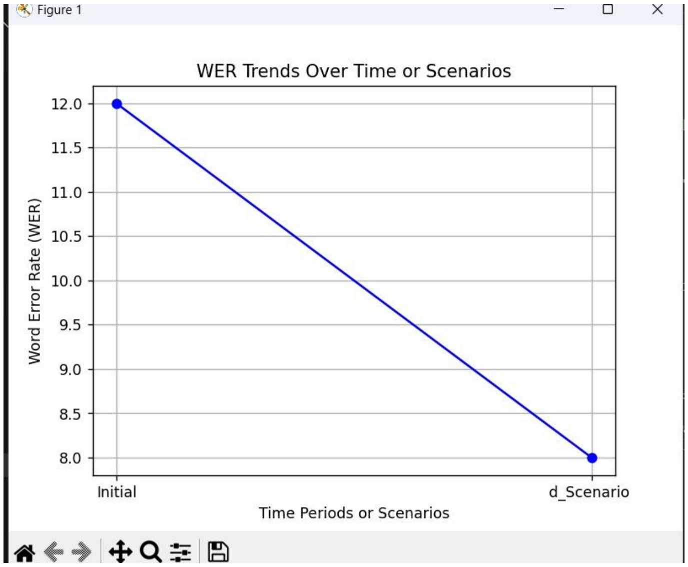
Y-axis: Accuracy or success rate



2. Word Error Rate (WER) Trends Over Time:

X-axis: Time period scenarios

Y-axis: Word Error Rate



Conclusion

In conclusion, the implemented multilingual lip synchronization system represents a significant achievement in the field of audio-visual processing. The project successfully addresses the challenges of synchronizing lip movements with spoken words across diverse languages.

By successfully integrating deep learning techniques, the system has demonstrated the capability to process and analyze both visual (lip movements) and auditory (spoken words) information concurrently.

The project sets a strong foundation for future research and applications in the realm of multimodal deep learning.