



EdZuban.AI

PROJECT SUPERVISOR

Dr. Jawwad Ahmad Shamsi

PROJECT TEAM

Warda Najam	K20-0310
Hira Asif	K20-0337
M. Minhal Manjee	K20-0467

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

FAST SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES KARACHI

CAMPUS

May 2024

Project Supervisor	Dr. Jawwad Ahmad Shamsi	
Project Team	Warda Najam	K20-0310
	Hira Asif	K20-0337
	M. Minhal Manjee	K20-0467
Submission Date	May 16, 2024	

Supervisor Name

Dr. Jawwad Ahmad Shamsi

Supervisor Sign



Dr. Zulfiqar Ali Memon

Head of Department

FAST SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES KARACHI CAMPUS

Acknowledgement

We express our profound gratitude to our FYP supervisor Dr. Jawwad Shamsi, Dean of the Computer Science department, for his invaluable contributions to the completion of our project, EdZuban.AI. We thank him for his time, efforts, and insightful advice throughout the year, as well as for his innovative ideas, solution suggestions, and weekly progress meetings. We also appreciate the support and guidance of our internal jury and FYP committee. This learning experience has been invaluable, and we are eternally grateful for this learning experience of a lifetime.

Abstract

Educational video tutorials offer a valuable resource for students, but they are often in English language and lack personalization and accessibility, especially for diverse learners or those seeking multilingual content. Additionally, current video translation and lip reanimation technologies have limitations in personalized concept suggestions.

Research Objective

This project aims to address these challenges listed above by developing an AI-powered system that integrates English language translation to Urdu, lip reanimation, and personalized learning functionalities into educational video tutorials.

Brief Methodology

- AI-based Video Translation (English to Urdu): Utilizing advanced machine translation models and audio-to-text alignment techniques, the system will translate educational videos into Urdu language, ensuring accuracy and fluency.
- Natural Lip Reanimation: LipGan is used to generate realistic lip movements synchronized with the translated speech, considering subtle facial expressions and emotions for enhanced naturalness.
- Personalization and Adaptive Learning: Utilizing user interaction data, the system will personalize the learning experience by recommending relevant concepts and providing targeted explanations.

Expected Benefits

- Improved Accessibility: Increased access to educational content for diverse learners and those seeking multilingual resources.
- Bridges the language gap: Urdu speakers gain access to a vast library of educational video resources, promoting inclusivity and knowledge sharing.
- Caters to diverse learners: Supports learners with different language backgrounds and learning styles, promoting equitable access to education.
- Natural and immersive experience: Lip reanimation and accurate translation foster deeper engagement and understanding of the learning material.
- Personalized learning paths: Recommendations and targeted explanations adapt to individual needs and preferences, leading to efficient knowledge acquisition and improved retention.
- Targeted support: The system provides timely and relevant explanations when needed, minimizing confusion, and maximizing learning time.
- Scalability and cost-effectiveness: The system can be adapted to various educational platforms and institutions, reaching a wider audience, and maximizing its impact.

Significance

This research has significant academic and industrial potential by promoting accessibility, personalized learning, and engagement within the educational technology domain. The developed system can revolutionize education by making learning more engaging, personalized, and effective for all learners. Additionally, the research will contribute to the advancement of AI-based video translation and lip reanimation technologies, with applications beyond the educational context.

Contents

Page

Introduction	9
Related Work	11
Requirements	18
Design	20
Implementation	28
Testing and Evaluation	36
Conclusion	50
References	51

LIST OF FIGURES

1	Fig1: Overall System Architecture of EdZuban.AI.....	17
2	Fig2: System Architecture view of EdZuban.AI.....	21
3	Fig3: Component Diagram.....	22
4	Fig4: EdZuban.AI UI.....	23
5	Fig5: Lip Reanimation UI.....	23
6	Fig6: State Transition Diagram.....	29
7	Fig7: Google Cloud Deployment.....	35
8	Fig8:Lip Reanimation UI.....	23

LIST OF TABLES

1. T1.....	24
2. T2.....	25
3. T3.....	26
4. T4.....	27
5. T5.....	36
6. T6.....	45
7. T7.....	45
8. T8.....	46
9. T9.....	47
10. T10	47
11. T11.....	49

1. Introduction

In Pakistan, a formidable educational challenge persists with more than 70% of students lacking English as their medium of instruction. This language barrier poses a significant obstacle, limiting their access to a wealth of online learning resources predominantly available in English. Prominent e-learning platforms like Coursera and Udemy primarily offer content exclusively in English, exacerbating the issue. However, recent advancements in artificial intelligence present a promising solution to address and mitigate this problem, thereby empowering a broader spectrum of learners in Pakistan to access quality education.

In the past, various apps and websites have offered translation services for videos in multiple languages, along with efforts in lip reanimation. However, there hasn't been a specific focus on Urdu, particularly within educational platforms. Personalization features have also been implemented on different platforms, but the integration of all these elements into a single educational platform is yet to be developed.

This project aims to harness the potential of machine learning to facilitate the accessibility of educational content for non-English-speaking students more specifically for Pakistani students to access education in their native language. Our approach focuses on three primary objectives:

- **Translation:**

We are developing a machine learning model capable of translating English-language educational videos into Urdu. This translation process ensures that students who are more comfortable with Urdu can fully comprehend and engage with the instructional material.

Translation has been done with the help of **OpenAI Whisper** and **Google Translate**.

- **Lip Reanimation:**

To enhance the immersive experience of learning, we are implementing lip reanimation techniques. By synchronizing the lip movements of video presenters with the translated Urdu audio, we aim to create a realistic illusion that the speaker is conversing in Urdu. This approach seeks to bridge the language gap and create a more engaging and inclusive online learning environment.

Lip Re-animation was carried out using **Wav2Lip** and **LipSync**.

- **Personalization:**

The objective of the personalization approach is to tailor the learning experience within video content by leveraging Natural Language Processing (NLP).

Personalization was done using the **Llama 2** model in order to generate explanations for technical terms.

Throughout the creation of this project, we encountered numerous constraints. While translating text was relatively straightforward, converting it to audio and then synchronizing it with the original video posed a significant challenge. This task required meticulous attention to detail to account for gaps, pauses, and to ensure the translated audio matched the original video's length.

Initially, lip reanimation using either wav2Lip or lipSync individually didn't produce highly accurate results. Thus, combining them proved essential to achieve the desired accuracy in the end product. While we managed to address these constraints during the development phase and have made improvements, there is still ample room for further enhancement.

Through the integration of these AI-driven solutions, this project aspires to contribute to the democratization of education in Pakistan, enabling a broader and more diverse audience to access high-quality learning resources. Ultimately, the successful implementation of this project could revolutionize online education accessibility and effectiveness, not only in Pakistan but also in regions facing similar language-based educational barriers.

2. Related Work

In this section, we provide an overview of previous studies conducted in this field and discuss the insights gained from them, which have inspired and informed our own research endeavors. Prajwal et al. (2020b) conducted a study focusing on the utilization of machine learning algorithms for lip-to-speech synthesis. Their work introduces a novel approach that considers individual speaking styles, leading to improved accuracy in synthesis. By leveraging audiovisual data, they train deep neural networks to capture distinctive lip movements and speaking styles, resulting in synthesized speech closely resembling the original. Their findings demonstrate that their method surpasses current techniques, yielding speech output akin to natural speech. K R et al. (2019) present a system designed for real-time translation of speech between individuals speaking different languages[1]. Their approach employs a multi-modal strategy, incorporating both audio and visual cues for translation. They introduce a pioneering visual module, LipGAN, which generates lifelike talking faces in real-time based on translated audio inputs. Their method exhibits superior performance compared to existing approaches, highlighting its potential for practical, real-time face-to-face translation applications. Ritter et al. (1999) explore the development of a translation agent capable of real-time face-to-face speech translation. Their research advocates for a multi-modal translation approach that integrates audio and visual data. By employing machine learning algorithms to analyze lip movements, speech patterns, and facial expressions of each speaker, they achieve real-time audio-visual translation with synchronized lip movement. Their results indicate the accuracy and potential applicability of their method in real-world scenarios. Regarding translation tools, Chitrlekha1 offers valuable functionalities by efficiently generating multilingual subtitles and voice-overs for informative videos. However, its efficiency may diminish when handling longer videos. Lastly, Huang et al. (2017) tackle the challenge of unpaired face translation between static images and dynamic videos, aiming to enhance video face prediction.]They propose a CycleGAN model with an identity-aware constraint, trained on a comprehensive face dataset and evaluated across various image and video scenarios[2]. Their findings demonstrate the effectiveness of their method in accurately translating faces between images and videos while preserving individual identities, outperforming existing approaches.

Educational video tutorials offer a valuable resource for students, but they are often in English language and lack personalization and accessibility, especially for diverse learners or those seeking multilingual content. Additionally, current video translation and lip reanimation technologies have limitations in personalized concept suggestions.

2.1. Objective

This project aims to address these challenges listed above by developing an AI-powered system that integrates English language translation to Urdu, lip reanimation, and personalized learning functionalities into educational video tutorials.

2.2 Brief Methodology

The proposed system integrates cutting-edge technology to enhance video translation, lip reanimation, and personalized learning experiences. Leveraging advanced machine translation models and audio-to-text alignment techniques, it facilitates the translation of educational videos from English to Urdu with a focus on accuracy and fluency. Additionally, the system employs LipGan technology to generate lifelike lip movements that synchronize seamlessly with the translated speech, capturing subtle facial expressions and emotions to enhance naturalness.[3] Furthermore, through the analysis of user interaction data, the system offers personalized learning experiences by recommending relevant concepts and delivering targeted explanations, thereby optimizing the learning journey for each user.

2.3 Expected Benefits

- Improved Accessibility: Increased access to educational content for diverse learners and those seeking multilingual resources.
- Enhanced Engagement and Learning Outcomes: Personalized and adaptive content will cater to individual needs and facilitate efficient knowledge gaining.
- Bridges the language gap: Urdu speakers gain access to a vast library of educational video resources, promoting inclusivity and knowledge sharing.
- Caters to diverse learners: Supports learners with different language backgrounds and learning styles, promoting equitable access to education.
- Natural and immersive experience: Lip reanimation and accurate translation foster deeper engagement and understanding of the learning material.
- Personalized learning paths: Recommendations and targeted explanations adapt to individual needs and preferences, leading to efficient knowledge acquisition and improved retention.
- Increased motivation and self-directed learning: Personalized learning promotes a sense of ownership and encourages autonomous learning habits.
- Reduced language barriers: Learners can access and understand educational content without needing to translate text or rely on subtitles.
- Targeted support: The system provides timely and relevant explanations when needed, minimizing confusion, and maximizing learning time.

2.4 Significance

This research has significant academic and industrial potential by promoting accessibility, personalized learning, and engagement within the educational technology domain. The developed system can revolutionize education by making learning more engaging, personalized, and effective for all learners. Additionally, the research will contribute to the advancement of AI-based video translation and lip reanimation technologies, with applications beyond the educational context.

2.5 Background and Justification

2.5.1 Background

Language barriers pose a significant challenge to accessing educational and informative video content. This issue too affects individuals who speak non-dominant languages or have limited language proficiency. Traditional video translation methods, such as subtitles or voice-overs, often lack naturalness and engagement, hindering learning and comprehension.

- Automated Subtitle Translation: This approach uses machine translation models to generate subtitles in the target language. However, these subtitles can be inaccurate, unnatural, and lack contextual understanding.
- Voice-over Dubbing: This technique involves recording a voice-over in the target language over the original audio track. While it can offer a more natural listening experience, it can be expensive and time-consuming to produce, and may not capture the original speaker's emotions and nuances.[19][20]
- Lip-syncing Avatars: This technology utilizes AI to create avatars that lip-sync to the translated audio. However, current approaches often struggle with natural facial expressions and movements, resulting in unnatural and distracting experiences[4].
- Lacking Personalization: There are no recommended explanations for linked concepts in a video.

Several AI-powered systems address video translation and learning, each with its own strengths and limitations:

- Subtitle Translation:
 - o Examples: Google Translate (2006), Microsoft Translator (2010), DeepL (2017)
 - o AI Technology: Machine translation models
 - o Features: Generate subtitles in target languages, often lacking naturalness and context.
- Voice-over Dubbing:
 - o Examples: Deepdub (2018), Resemble AI (2019)
 - o AI Technology: Speech synthesis, text-to-speech models
 - o Features: Offers a natural listening experience but can be expensive, time-consuming, and lacks emotion capture.
- Lip-syncing Avatars:
 - o Examples: Animaze (2016), Synthesia (2018)
 - o AI Technology: Generative adversarial networks (GANs)
 - o Features: Create avatars that lip-sync to translated audio, often struggling with natural facial expressions.
- Personalized Learning Platforms:
 - o Examples: Khan Academy (2006), Duolingo (2011), Coursera (2012)
 - o AI Technology: Recommendation systems, adaptive learning algorithms.
 - o Features: Recommend relevant content and adapt difficulty based on user interaction, often lacking personalized explanations and integration with translated content.

2.5.2 Justification:

While existing solutions offer some benefits, they fall short in providing a truly natural, engaging, and personalized learning experience for Urdu speakers all in one. This project seeks to address these limitations by developing an AI-powered system that specifically targets English-to-Urdu translation, integrates it with natural lip reanimation, and personalizes the learning journey.

2.6 Enhanced System Features:

- English-to-Urdu Translation: Use advanced MT models trained on large datasets of diverse content to ensure high accuracy and fluency.
- Natural and Expressive Lip Reanimation: Employ LipGan to generate realistic and synchronized lip movements that reflect the translated content.
- Personalized Learning and Explanation: Analyze user interaction data and learning preferences to recommend relevant concepts, adapt difficulty, and provide targeted explanations in Urdu for improved understanding and knowledge retention.

2.7 Justification for Focus on Urdu

- Large Urdu-speaking population: Over 250 million global speakers, primarily concentrated in India and Pakistan.
- Limited access to translated resources: Scarcity of translated educational and informative video content in Urdu.
- Bridging the language gap: This project can significantly improve access to knowledge and information, fostering educational equity and cultural understanding.

2.8 Contribution to Existing Research

- Culturally-aware translation and lip reanimation: Cater to specific nuances of Urdu language and lip expressions for an authentic user experience.
- Integrated personalized learning: Combine translation and lip reanimation with adaptive recommendations and targeted explanations for effective knowledge acquisition.
- Focus on underserved population: Address the needs of Urdu speakers who often lack access to culturally relevant educational resources.

2.9 Problem Statement

2.9.1 Language Barrier and Limited Accessibility of Educational Content

Most of the educational and informative video content is available primarily in English, creating significant barriers for non-English speakers. This is particularly true for Urdu speakers, who represent a substantial global population with limited access to translated resources. This lack of access hinders learning opportunities, cultural awareness, and educational equity.

2.9.2 Challenges of Existing Translation and Learning Systems

- Unnatural and Inaccurate Translation: Voice actors or models may produce inaccurate or unnatural-sounding translations, hindering comprehension and engagement.
- Lack of Lip Reanimation: Subtitles and voice-overs often lack synchronized lip movements, creating a distracting and unnatural viewing experience.
- Limited Personalization: Many systems fail to tailor the learning experience to individual needs and preferences, leading to decreased motivation and suboptimal learning outcomes.
- Focus on Major Languages: Existing systems often prioritize English translation to major languages, neglecting languages like Urdu that have a significant number of speakers but limited access to translated resources.

2.9.3 Research Challenges and Risks

- Achieving High-quality English-to-Urdu Translation: Accurately translating complex content while maintaining idiomatic usage and cultural nuances in Urdu poses a significant challenge.
- Natural and Expressive Lip Reanimation: Generating synchronized and natural lip movements that reflect the translated Urdu speech.
- Personalization and Adaptive Learning: Effectively shaping the learning experience to individual preferences and learning styles necessitates robust user interaction analysis and adaptation algorithms.
- Data Availability and Bias: Accessing sufficient and diverse techniques and methods, while mitigating potential biases in the data is equally important.

2.10 Language Translation in Videos

2.10.1 Automatic Speech Recognition (ASR)

ASR forms the foundation of video translation, converting the audio track into a textual representation for subsequent language processing. Recent advancements in deep learning have resulted in significant progress in ASR performance, particularly with models like DeepSpeech2 and Wav2Vec2 achieving high accuracy for various languages and domains[5][6].

2.10.2 Machine Translation (MT)

MT algorithms are responsible for translating the extracted text from the audio track into the desired target language. Transformer-based models like Google Translate and T5 have demonstrated remarkable fluency and accuracy in text translation.

2.10.3 Audio-to-Text Alignment

- Phoneme-based alignment: Aligns translated text with phonemes extracted from the original speech.
- Speaker-specific models: Leverages large datasets of audio-video data for specific speakers to personalize lip sync accuracy.

2.10.4 Lip Reanimation

- Parametric models: Control facial movements through pre-defined parameters.
- Deep learning models: Employ neural networks to learn the relationship between audio and lip movements.
- Physics-based models: Utilize physical simulations to create realistic facial deformations.

2.11 Personalized Learning and Adaptive Systems:

- Knowledge tagging: Annotating video content with relevant tags based on concepts and learning objectives.
- Interactive overlays: Presenting pop-up explanations, quizzes, and additional resources related to specific concepts within the video.
- Branching narratives: Providing alternative video paths based on user responses and knowledge gaps.
- Adaptive recommendations: Suggesting related video tutorials based on the user's learning history and interests.
- Personalization profiles: Allowing users to customize the video playback speed, subtitles, and other interface features.
- User modeling: Building user profiles that capture learning preferences, knowledge levels, and performance data.
- Concept identification: Automatically identifying and tagging relevant concepts within video content[7].
- Adaptive content generation: Generating personalized explanations, recommendations, and learning pathways.
- Real-time feedback: Providing immediate feedback on user interactions and progress.

2.12 Existing Research and Projects:

- Khan Academy: Adaptively recommends practice problems and exercises based on user performance.
- Edmodo: Provides personalized learning paths and playlists based on user interests and goals.
- Adapt Learning: Implements AI-powered video tutorials with personalized recommendations and quizzes.
- Udacity: Offers adaptive nanodegrees that automatically adjust the learning path based on user progress.

2.13 Methodology

2.13.1 Translation and Lip Reanimation

- Video Transcription: Whisper, an open-source automatic speech recognition (ASR) model, is used to transcribe the audio track of the video from English to text.
- Machine Translation: Google Translate API is employed to translate the transcribed English text into Urdu.

- Text-to-Speech Conversion: Azure Cognitive Speech Services API converts the translated Urdu text into natural-sounding speech.
- Lip Reanimation: EdZuban.AI leverages existing lip reanimation APIs that map the synthesized Urdu speech to the video speaker's facial movements, creating natural and synchronized lip movements[8].

2.13.2 Personalization

- User Interaction Analysis: User interactions are monitored and analyzed to identify learning preferences and knowledge gaps.
- Personalized Recommendation: Based on the user's learning history and performance, EdZuban.AI recommends relevant concepts for further explanation and personalized learning paths[9].

2.13.3 System Architecture

- Input Module: Handles video input, extracts audio, and transcribes it to English text.
- Translation Module: Utilizes Google Translate API for English-to-Urdu translation.
- Speech Synthesis Module: Employs Google Text-to-Speech API to convert Urdu text to speech[10].
- Lip Reanimation Module: Integrates existing APIs for natural and synchronized lip movements.
- Personalization Module: Analyzes user interaction data and generates personalized recommendations and learning paths.
- Output Module: Delivers the translated video with natural lip reanimation and personalized learning elements.

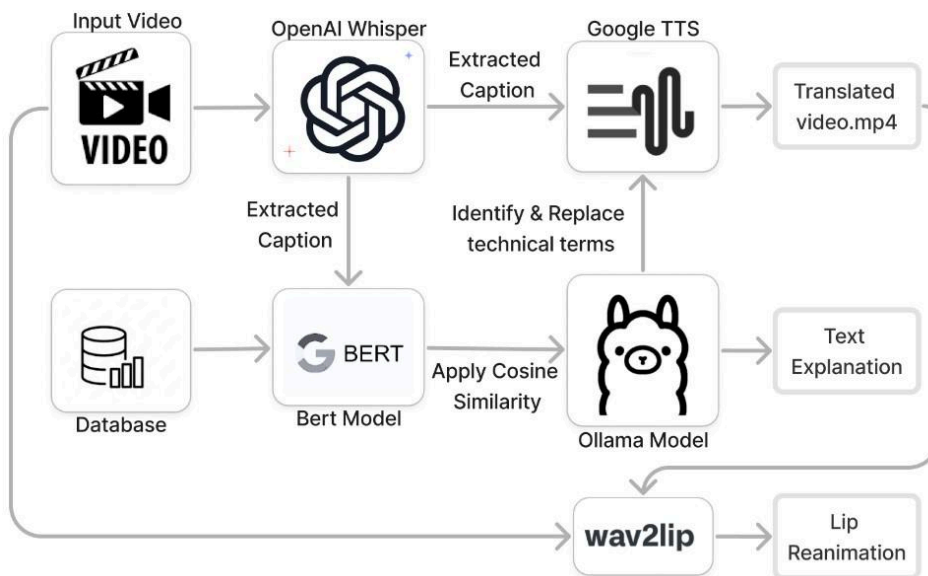


Fig 1. Overall System Architecture of EdZuban.AI

3 Requirements

The overall system has minimal requirements, necessitating only the correct system configuration and the installation of specific Python libraries.

3.1 Operating System Compatibility:

The software is compatible with the following versions of the Microsoft Windows operating system:

- Windows 10 (64-bit) version 1809 or later
- Windows 8.1 (64-bit)
- Windows 7 SP1 (64-bit)

3.2 Hardware Requirements:

Minimum hardware requirements for running the software are as follows:

- Processor: Intel Core i3 or AMD equivalent
- RAM: 4GB
- Storage: 500MB free disk space
-

3.3 Software Dependencies:

The software requires the following dependencies to be installed on the system:

- Python 3.11.3 or later
- Visual C++ Redistributable for Visual Studio 2015 or later
-

3.4 Library dependencies

This project necessitates the following software libraries to function effectively:

- Flask==2.3.2
- Flask-Cors==4.0.0
- nltk==3.8.1
- google-ai-generativelanguage==0.6.2
- ipython==8.11.0
- googletrans==4.0.0rc1
- pydub==0.25.1
- soundfile==0.12.1
- librosa==0.10.1
- SpeechRecognition==3.10.3
- moviepy==1.0.3
- gTTS==2.4.0
- openai-whisper==20230918

3.5 Deployment Requirement:

To deploy this project, you must have an active account on the Google Cloud Portal with payment methods already configured. Initially, the project has been deployed using a student account; however, a transition to a different account may be necessary in the future.

3.6 Internet Connection:

An active internet connection is required for initial software activation and call to apis.

3.7 User Permissions:

Administrator privileges might be required for installation and certain features of the software.

4 Design

Major components that the software architecture of this project can be divided into:

1. Video Translation Module
 2. Lip Re-animation Module
 3. Personalization Module
 4. User Interface Module
- **Relationships and Dependencies:**
Interdependencies exist among these modules. The translation module initially takes a video input for conversion into Urdu. Through various APIs, both captions and audio within the video are translated and synchronized. Following this, the video moves to the lip re-animation module, which adjusts the speaker's lip movements to match the translated language. Subsequently, the video enters the personalization module, where it's monitored for user interactions. If the user clicks on an unfamiliar term within the captions, the system provides an explanatory response, enhancing user understanding on various subjects.[11]
All these processes occur in the background, integrated with the User Interface Module. This front-end component allows users to input videos and view the translated content seamlessly.
 - **Interfaces with External Systems:**
The internal interface will connect and interact with the external interface, encompassing:
 1. **Translation APIs:** Google Translation API and Azure Cognitive Services are integrated to facilitate language translation, ensuring robust conversion capabilities.
 2. **NLP Libraries:** Models for accurate and contextually relevant translations are incorporated into the system.
 3. **Wav2Lip API:** Interaction with the Wav2Lip API enables customized lip synchronization aligned with the user's native language.

4.1 Software Architecture

- **User Interface Layer:** It will be presenting information to users and accepting user inputs like videos that they would want to be translated in Urdu.
- **Middle Tier / Business Logic Layer:** It contains the application's business logic, handling processes and data manipulation like connecting to APIs that will be helping in translating the videos, generating captions and lip reanimation.

- **Data Access Layer:** This layer will manage the actual retrieval and storage of data. It will interact directly with the database, so when a user clicks on an unfamiliar term, its explanation can be retrieved from there. Also if the user has studied that topic previously or not can also be stored in a csv file to keep a track of it.

This layered architecture demonstrates the separation of concerns and responsibilities among different layers, allowing for better scalability, maintainability, and reusability of code components.

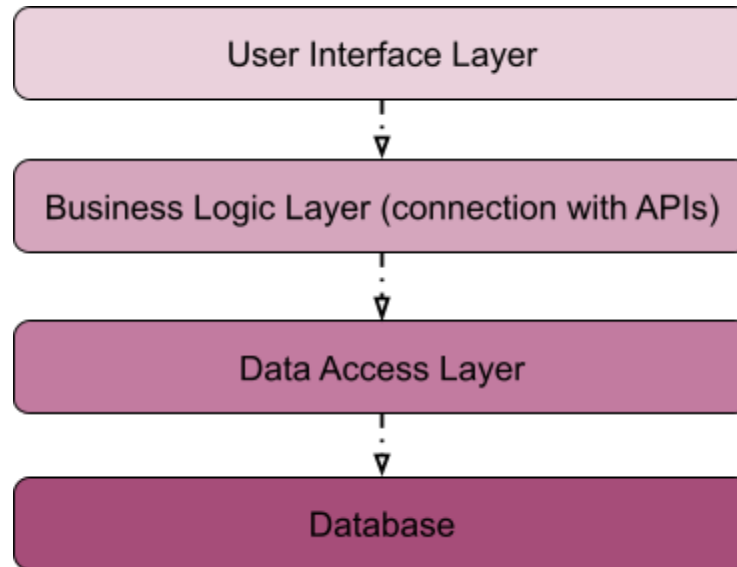


Fig 2. System Architecture view of EdZuban.AI

4.2 Design Strategy

The current design strategy involves modularization, where each method functions as an independent component collaborating with others. Starting from the UI, users input videos for translation, processed through the Open AI Whisper API for transcription and Urdu conversion. This translated text moves to another component for text-to-speech and video synchronization, generating a Translated_video.mp4. For further precision, it undergoes lip-syncing via the Lip GAN and Wav2Lip API to align with the new language.

In parallel, the personalization system checks for unfamiliar words in the database. If found, it provides explanations; otherwise, it offers a general response. This integration ensures users receive translated content seamlessly via the frontend.

The system architecture is depicted in the component diagram below:

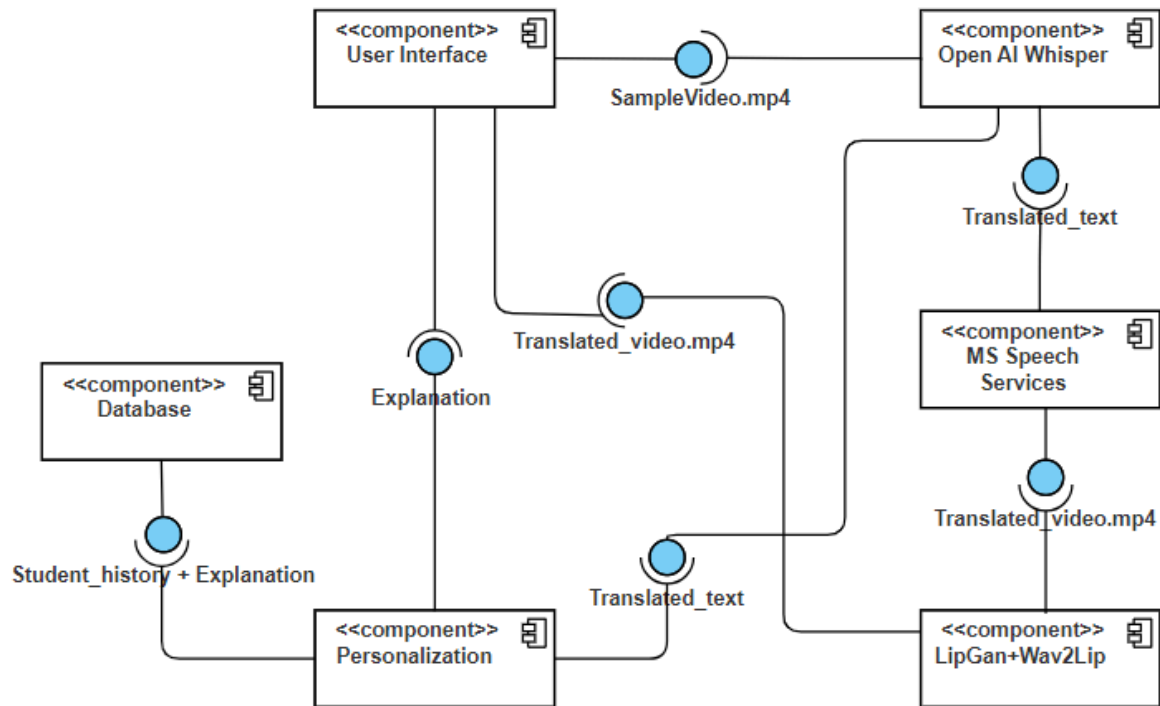


Fig 3.Component Diagram

4.2.1 System Reuse: The system exhibits reusability across several modules designed to promote versatility and potential reuse in future endeavors:

Translation Modules: The translation modules integrated into the system, such as the Open AI Whisper API for transcription and language conversion, serve as reusable components. These APIs possess the flexibility to be repurposed for various language translations beyond the current English to Urdu capability.[18]

Text-to-Speech and Synchronization Components: The system's text-to-speech and video synchronization modules, working in conjunction to create synchronized translated videos, stand as reusable components. They can be leveraged for synchronization processes across different languages and content types[12][13].

Lip Reanimation and Video Enhancement: The Lip GAN and Wav2Lip APIs, employed for lip reanimation aligned with the translated language, serve as adaptable components. Their capabilities can be utilized for similar video enhancement tasks in multiple language translations[17].

Strategies deployed to facilitate reusability encompass the use of standardized APIs and libraries, allowing seamless integration into different segments of the system or potential future projects. Additionally, the modular approach and component-based architecture inherently promote the reusability of specific functionalities across various language translation processes.

4.2.3 User Interface:

Explanation

Social enterprise: A business that operates with a social mission and reinvests its profits into the community.

Scratch: A coding platform for children that allows them to create interactive stories, games, and animations.

Web: The interconnected network of websites and pages accessible on the internet.

Mobile: Relating to devices such as smartphones and tablets that can be carried around and used on the go.

Video: Recorded moving images and sound that can be watched on a screen.

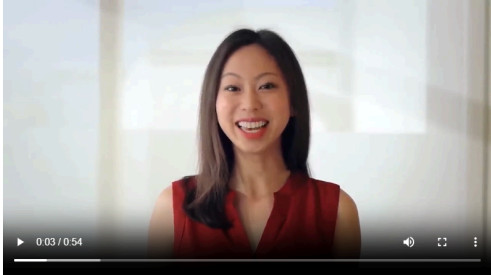
Blue-collar worker: A person who performs manual labor in a non-office environment, such as a factory worker or construction worker.

Technology: The application of scientific knowledge for practical purposes, especially in industry or commerce.

Silicon Valley values: A set of beliefs and principles that are common in the

Deploy ⋮

Video: 5.mp4



Personalized Glossary

Hi everyone, my name is Ying-Ying Liu and I'm a 26-year-old American from Atlanta, Georgia. I've recently arrived in the Bay Area after spending the past three and a half years in China. When 2012, I co-founded an award-winning social enterprise through scratch. Using web, mobile and video, my team and I placed hundreds of blue collar workers in good jobs. Today, this company is self-sustaining. I saw with my own eyes the power that technology has to accelerate impact. This is why I'm now here because Silicon Valley values smart risk-taking, innovation and ruthless efficiency. I'm looking for a company that's mission-inspired and out to change the world because I am too.

Fig 4. EdZuban.AI UI

Video

Drop File Here
- OR -
Click to Upload

Audio

Drop File Here
- OR -
Click to Upload

Checkpoint
Name of saved checkpoint to load weights from

☐ Translip-1 ☐ Translip+gan

Prevent smoothing face detections over a short temporal window

☐ No Smooth

Resize Factor
Adjust Resolution

1

Pad Top
Padding above lips

0

Pad Bottom (20 and above will include chin)
Padding below lips

10


Pad Left
Padding to the left of lips

0

Pad Right
Padding to the right of lips

0

Video



Generate TransLip Magic

Fig 5. Lip Reanimation UI

4.2.4 Language Translation Use Cases

- **UC-1.1 (Educational Video Translation):**
 - Description: A user, an educator, wants to translate an English educational video into Urdu for their students who are more comfortable with that language.
- **UC-1.2 (Cross Cultural Communication in Offices):**
 - Description: A professional working in an international company needs to understand a meeting recorded in English but prefers it in Urdu for better comprehension.

4.2.5 Site Personalization

- **UC-2.1 (Language Learning Assistance):**
 - Description: A language learner watches a foreign language video to improve their vocabulary and comprehension.
- **UC-2.2 (Technical Jargon Clarification):**
 - Description: A professional in a technical field views a translated instructional video with specific jargon they're unfamiliar with.
 - Description: A professional in a technical field views a translated instructional video with specific jargon they're unfamiliar with.

Table 1. UC-1.1: Educational Video Translation

UC-1.1: Educational Video Translation		
Use case Id:	UC-1.1	
Actors:	Language Learner initiated, APIs, System	
Feature:	UR-1 (Translation of the video)	
Pre-condition:	The uploaded video is in English. Language preferences are set to English-to-Urdu translation. The ML framework is connected to translation APIs. Translation is successful, so that captions are shown.	
Scenario: The user uploads the English video onto the platform and selects Urdu as the preferred translated language. The system translates the video and generates synchronized captions, enabling a seamless learning experience.		
Step#	Action	Software Reaction
1.	Uploads Video	
2.	Selects urdu	UI sends to OpenAI Whisper API to transcribe
3.		Translated text to speech using MS Speech Services
4.		Performs Lip reanimation
		UI displays the Translated Video
Alternate Scenarios: NA		
1a:		
2a:		
Post Conditions		
Step#	Description	
	The translated video with synchronized captions is available for the educator and students to access on the platform.	
Use Case Cross referenced	<Related use cases, which use or are used by this use case>	

Table 2. UC-1.2: Cross Cultural Communication in Offices

UC-1.2: Cross Cultural Communication in Offices		
Use case Id:	UC-1.2	
Actors:	Professional Office Worker, API, System	
Feature:	UR-2 (Turn on/off captions)	
Pre-condition:	The translated video with synchronized captions is available and accessible. The user is watching the translated video with active captions. The system has a glossary or a database containing explanations for unfamiliar terms.	
Scenario: The user inputs the English video, selects Urdu translation, and turns on the translated captions. The system efficiently translates the dialogue into Urdu, aiding the user's understanding of the meeting's content.		
Step#	Action	Software Reaction
1.	Office worker uploads video	
2.	Selects urdu	UI sends to OpenAI Whisper API to transcribe
3.	Turns on captions	UI Displays Captions in urdu
4.		UI forms a text file that can be downloaded with urdu notes
Alternate Scenarios: NA		
1a:		
2a:		
Post Conditions		
Step#	Description	
	The user gains a better understanding of the previously unfamiliar term through the displayed explanation or glossary entry.	
	The video playback remains uninterrupted, continuing from where the user clicked the unfamiliar term.	
Use Case Cross referenced	<Related use cases, which use or are used by this use case>	

Table 4. UC-2.2: Technical Jargon Clarification

UC-2.2: Technical Jargon Clarification		
Use case Id:	UC-2.2	
Actors:	User, System	
Feature:	UR-4 (Click on unfamiliar words)	
Pre-condition:	<List the assumptions required before this Use Case can be executed. >	
Scenarios As the user watches the translated content, they click on technical terms within the captions. The system presents explanations or glossary entries relevant to the clicked terms, enhancing the user's understanding of the subject matter.		
Step#	Action	Software Reaction
1.	User clicks on unfamiliar word	Searches database for technical words
2.		Displays explanation
Alternate Scenarios:		
1a: The user clicks on an unknown term. The system doesn't have an entry for the clicked term in its glossary or database. The system displays a message indicating that no explanation is available for the selected term.		
2a:		
Post Conditions		
Step#	Description	
	The user gains a better understanding of the previously unfamiliar term through the displayed explanation or glossary entry.	
	The video playback remains uninterrupted, continuing from where the user clicked the unfamiliar term.	
Use Case Cross referenced	<Related use cases, which use or are used by this use case>	

5 Implementation

There are a total of 10 states the program is working on:

1. **Video Received:** The program initiates when it receives a video file as input for translation.
2. **Check API Availability:** Before processing, the system verifies the availability and accessibility of necessary APIs required for translation and synchronization.
3. **Check Video Constraint:** The system checks the video file to ensure it meets the required constraints or specifications necessary for processing and translation.
4. **Caption Generated:** Upon successful verification of the video, the program generates captions based on the video content.
5. **Translated Audio:** Simultaneously, it translates the audio content from the original language to the desired target language.
6. **Video Synchronized:** The program synchronizes the translated audio with the video, ensuring alignment and coherence between audio and visual elements[26].
7. **Video Reanimated:** If required, the system processes the video through reanimation processes, focusing on lip synchronization based on the translated audio[27].
8. **Personalization:** The program enters a phase where it monitors user interactions. It generates explanations of unfamiliar terms within the captions.[14]
9. **User Database:** A database containing user-related information, preferences, or history to personalize the user's experience or interaction with the translated content.
10. **Word Database:** This database contains an array of terms, explanations, or glossary entries. It's accessed when users seek explanations for unfamiliar terms within the translated captions.

These states outline the sequence of steps the program undergoes, now including a check for API availability to ensure that the necessary tools for translation and synchronization are accessible before processing the video. The state diagram describes the process below.

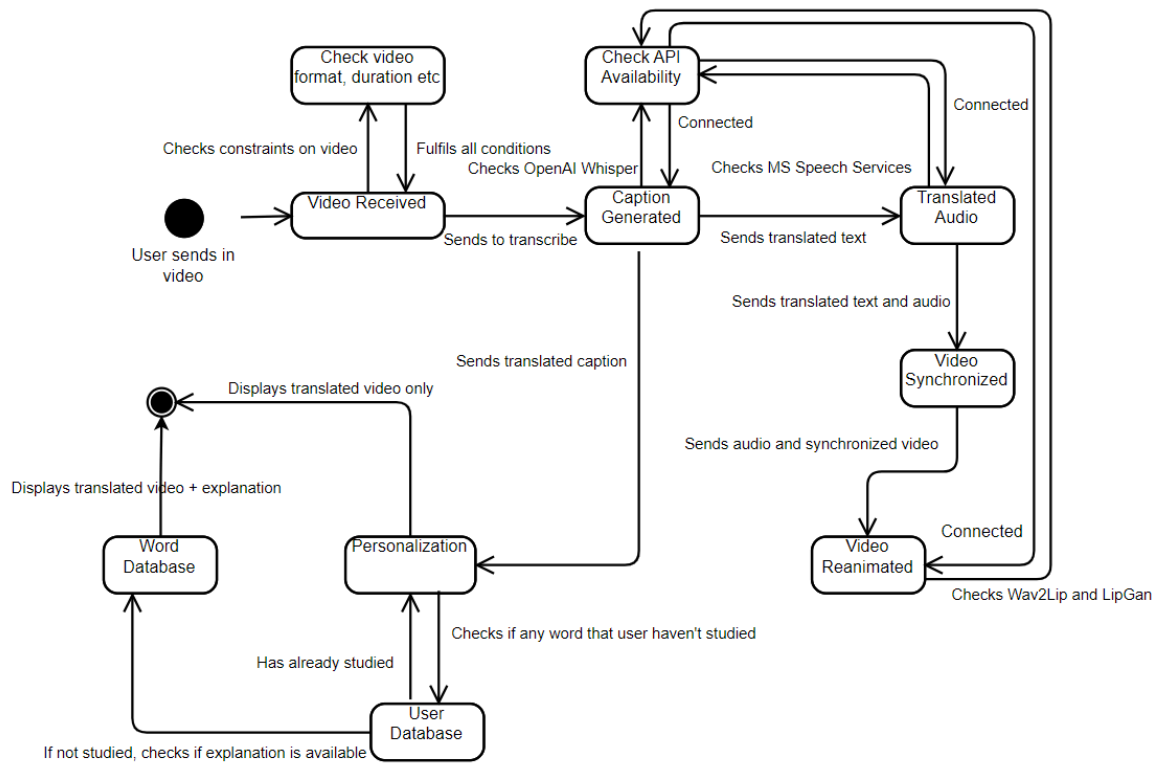


Fig 6. State Transition Diagram

5.1 Methodology - Coding Implementation

Our approach involves the depiction of real time translation so we have created chunks, each of 15s. While a chunk is being displayed another chunk is processed.

5.2 Libraries being used:

```

import moviepy.editor as mp
from googletrans import Translator
from gtts import gTTS
import os
import whisper

```

5.3 Load video:

```

def process_video(video_path, chunk_size=15):
    clip = mp.VideoFileClip(video_path)
    duration = clip.duration
    total_duration = 0

```

5.4 Audio Extraction:

To begin, we will extract audio from the video using Python's MoviePy Library, a versatile tool designed for various video editing tasks. This will happen for each chunk size.

```
for i in range(0, int(duration), chunk_size):
    start_time = i
    end_time = min(i + chunk_size, duration)
    sub_clip = clip.subclip(start_time, end_time)

    # Extract audio from the chunk and convert to text
    audio_path = f"./temp_{i}.wav"
    sub_clip.audio.write_audiofile(audio_path)
```

5.6 Transcribing Audio:

After successfully extracting the audio, our next step involves transcribing the spoken content into text in the English language. To accomplish this, we will utilize OpenAI's Whisper, an advanced automatic speech recognition system. Whisper boasts its impressive accuracy, which surpasses that of other models available.

Whisper's effectiveness is underpinned by its architecture, which is based on a vast dataset comprising 680,000 hours of multilingual and multitasking data sourced from the internet. This architecture employs a straightforward end-to-end approach, implemented as an encoder-decoder Transformer. The process begins by segmenting the input audio into 30-second chunks, followed by conversion into a log-Mel spectrogram. Subsequently, these transformed audio segments are fed into an encoder. To complete the process, a decoder is trained to predict the corresponding text captions for the audio.

The selection of Whisper for this task is driven by its remarkable accuracy, making it proficient in generating accurate and reliable transcripts from audio content. This ensures that the transcriptions are of the highest quality and fidelity to the spoken speech.

```
def audio_to_text(audio_file, whisper_model="base"):
    model = whisper.load_model('base.en')
    option = whisper.DecodingOptions(language='en', fp16=False)
    result = model.transcribe(audio_file)
    return result['text']
```

5.7 Translation of Script to Urdu:

This text is then being translated to urdu using Google Translation API and is stored in *translated_text* variable

```
def translate(text, dest='ur'):
    translator = Translator()
    translated = translator.translate(text, dest=dest)
    return translated.text
```

5.8 Urdu to Audio Generation:

Then Google Text-to-Speech is being used to convert the translated text into an audio file. For this we just need to input the urdu transcript in the Text-to-Speech system. Many TTS systems allow us to choose from different voices and adjust parameters like speaking rate, pitch, and volume. We can configure these settings according to your preferences. Then we will initiate the TTS system to generate the Urdu audio from the transcript[22]. The system will process the text and produce an audio file in the Urdu language.

```
tts = gTTS(text=translated_text, lang='ur')
translated_audio_path = f"./translated_audio_{i}.mp3"
tts.save(translated_audio_path)
```

5.9 Modifying original video:

This step involves the overlaying of translated audio on the original video chunk replacing its original audio that was in english.

```
translated_audio_clip = mp.AudioFileClip(translated_audio_path)
sub_clip = sub_clip.set_audio(translated_audio_clip)

total_duration += chunk_size
output_video_path = f".translated_video_{total_duration}.mp4"
sub_clip.write_videofile(output_video_path, audio_codec='aac')

# Clean up temporary files
os.remove(audio_path)
os.remove(translated_audio_path)

yield output_video_path
```

5.10 Personalization of Captions:

To start working with personalization, we will pass the english caption of the video that was being generated through OpenAI Whisper API[16]. This text will be first preprocessed inorder to remove punctuation, stop words and non-alphabetic characters.

```
def preprocess_text(text):
    # Tokenization
    tokens = word_tokenize(text.lower()) # Convert to lowercase

    # Remove punctuation and non-alphabetic characters
    tokens = [word for word in tokens if word.isalpha()]

    # Remove stopwords
    stop_words = set(stopwords.words('english'))
    tokens = [word for word in tokens if word not in stop_words]

    # Lemmatization
    lemmatizer = WordNetLemmatizer()
    tokens = [lemmatizer.lemmatize(word) for word in tokens]

    return tokens
```

5.11 Extract Keywords/Tokens used in the caption:

Make a dictionary of the preprocessed text and build a Latent Dirichlet Allocation model on it. LDA assigns weight to each word and extracts important keywords/ tokens from it.

```
# Create a dictionary representation of the documents
dictionary = corpora.Dictionary(preprocessed_texts)

# Convert tokenized documents into a document-term matrix
corpus = [dictionary.doc2bow(text) for text in preprocessed_texts]

# Build LDA model
lda_model = models.LdaModel(corpus, num_topics=3, id2word=dictionary,
passes=15)

# Print topics
for idx, topic in lda_model.print_topics():
    print(f'Topic {idx + 1}: {topic}')

# Extracted topics from LDA model
extracted_topics = [lda_model.print_topic(i) for i in range(lda_model.num_topics)]

# Formatting topics into a single string
formatted_topics = "\n".join(extracted_topics)
```


5.12 Extract Complete Topics:

Using the Llama 2 model and passing the relevant prompt we will extract the complete topic names from the weighted tokens produced by the LDA model.

```
from llama import LLAMATemplate, LLAMACHain
llama = LLAMA("2.7b-chat")

# Define template for generating prompts
template = ""Based on the tokens you will be provided, you have to suggest
relevant topic names.
Question: {extracted_topics}
Answer:
""

llama_template = LLAMATemplate(input_variables=["extracted_topics"],
template=template)
llama_chain = LLAMACHain(llama=llama, template=llama_template)
answer = llama_chain.run(formatted_topics) # Using the formatted_topics as input
```

5.13 Similarity between Extracted Topics and Studied Topics:

The BERT model (bert-base-nli-mean-tokens) was employed to compute embeddings for both the extracted topics from the English text and the topics the student has already covered. Subsequently, the similarity between these two sets of embeddings was determined using the Cosine Similarity function. If there are topics in the English text that the user hasn't studied yet, those topics are recommended for generating further explanation.

```
model = SentenceTransformer('bert-base-nli-mean-tokens')
excel_file = 'student_info.xlsx'
df = pd.read_excel(excel_file)
not_studied_topics = df[df['Studied'] == 0]['Topic'].tolist()

suggested_topic_embeddings = model.encode(suggested_topics)
not_studied_topic_embeddings = model.encode(not_studied_topics)

similarity_matrix = cosine_similarity(suggested_topic_embeddings,
not_studied_topic_embeddings)
for i, topic_similarities in enumerate(similarity_matrix):
    similar_topics_indices = topic_similarities.argsort()[::-1] # Sort by similarity
    similar_topics = [not_studied_topics[idx] for idx in similar_topics_indices]

preprocessed_text = preprocess_text(caption_text)
```

5.14 Generate Explanation for Technical Terms:

Once again using Llama 2, we extract the technical terms used in the text from the topic that the student is unfamiliar with.

```
from llama import LLAMATemplate, LLAMACHain

llama = LLAMA("2.7b-chat")

template = """Identify any technical terms related to the given topic:
Context: {text}
Technical Terms:
"""

llama_template = LLAMATemplate(input_variables=["preprocessed_text"],
template=template)
llama_chain = LLAMACHain(llama=llama, template=llama_template)

answer = llama_chain.run(formatted_text)
```

5.15 Explanation Generation:

Generate explanation of technical terms using wordnet, if there is no explanation for a term there then Llama 2 can be used again to give explanation keeping the context in mind.

```
def get_word_definition(word):
    synsets = wordnet.synsets(word)
    if not synsets:
        return f"No definition found for '{word}'"
    return synsets[0].definition() if synsets else f"No definition found for '{word}'"
```

This completes the video translation and personalization process.

5.16 Lip Re-animation with generated Audio:

Once the Urdu audio is generated, we can proceed to work with the original video, now stripped of its initial audio. We will use this video alongside the newly created Urdu audio as inputs for an API called Wav2Lip. This API, pretrained on the LRS2 dataset, is specifically designed for lip reanimation tasks.

Furthermore, to enhance the capabilities of the Wav2Lip API and improve its performance, we have the option to incorporate additional datasets containing audio and video samples. These supplementary datasets can serve to further train and fine-tune the API, allowing it to produce more accurate and contextually aligned lip animations.

In summary, this workflow involves pairing the Urdu audio with the video, leveraging the Wav2Lip API pretrained on the LRS2 dataset, and optionally incorporating additional datasets to optimize lip reanimation results. This comprehensive approach aims to create a seamless and engaging viewing experience, where the lip movements align precisely with the spoken Urdu audio

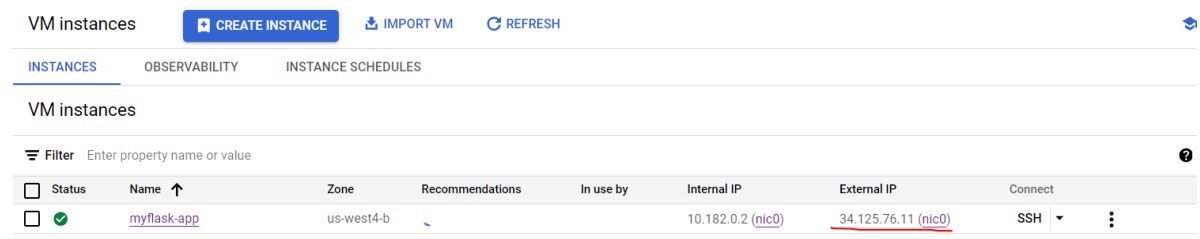
5.17 Deployment on Google Cloud Portal:

1. Create a virtual machine instance and form your flask application
2. Access it by connecting ssh

Code:

```
Sudo su
python3 -m venv venv
source venv/bin/activate
nano app.py [complete all code files like this]
pip install googletrans [install all relevant libs]
python3 app.py
```

3.



VM instances							
INSTANCES OBSERVABILITY INSTANCE SCHEDULES							
VM instances							
Filter Enter property name or value							
Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
<input checked="" type="checkbox"/>	myflask-app	us-west4-b			10.182.0.2 (nic0)	34.125.76.11 (nic0)	SSH

Fig 7. Google Cloud Deployment

Using this external IP, we can call our endpoints using this IP like:

http://34.125.76.11:8080/get_chunk

<http://34.125.76.11:8080/personalization>

6 Testing and Evaluation

Different types of testing and evaluation metrics were used to gauge the effectiveness of our project.

6.1 Test cases covering black box testing

Table 5. Test Cases

Feature	Test Case ID	Test Case Description	Input Data	Expected Result	Actual Result	Status
Machine Translation	MT001	Translate educational video on mathematics from English to Urdu	Mathematics video (10 mins)	Accurate Urdu translation preserving context and meaning	Accurate translation with minor context adjustments needed	Pass
	MT002	Translate educational video on history from English to Urdu	History video (15 mins)	Accurate Urdu translation preserving context and meaning	Accurate translation with some historical terms needing manual intervention	Pass
	MT003	Translate technical terms accurately in a computer science video	Computer science video (20 mins)	Technical terms accurately translated to Urdu	Technical terms mostly accurate, minor inaccuracies noted	Pass
	MT004	Handle everyday expressions in an English video	English video (12 mins)	Casual expressions accurately translated to Urdu	Casual expressions translated, some slightly lost	Pass
Lip Synchronization	LS001	Synchronize lip movements for a short video on science	Science video (3 mins)	Translated Urdu audio perfectly synchronized with original speaker's lip movements	Smooth synchronization with minor delays	Pass

	LS002	Synchronize lip movements for a long lecture on economics	Economics lecture (30 mins)	Translated Urdu audio perfectly synchronized with original speaker's lip movements	Synchronization successful with occasional mismatch during fast speech	Pass
	LS003	Evaluate synchronization quality for videos with fast-paced speech	General Tutorial(10 mins)	Smooth synchronization without noticeable delays or mismatches	Minor desynchronization during rapid speech sections	Pass
	LS004	Test synchronization for videos with multiple speakers	Panel discussion video (20 mins)	Accurate synchronization for all speakers, maintaining lip-sync quality		To do
Personalized Learning	PL001	Generate explanations for technical terms in a physics video	Physics video (15 mins)	Clear and accurate explanations for technical terms	Clear explanations	Pass
	PL002	Generate context-sensitive explanations for complex concepts in a chemistry video	Chemistry video (10 mins)	Explanations tailored to context, improving understanding of complex concepts	Context-sensitive explanations generally accurate, some improvements needed	Pass
	PL003	Provide personalized examples based on a biology video	Biology video (8 mins)	Relevant examples that enhance understanding, considering user's prior knowledge		To do

	PL004	Adapt explanations dynamically based on user feedback	Interactive video (any subject, 10 mins)	Adjusted explanations based on real-time user feedback, improving comprehension and engagement		To do
Overall Integration	INT001	End-to-end test: Translate, synchronize , and personalize learning for a general knowledge video	General knowledge video (5 mins)	Seamless integration of translation, lip synchronization, and personalized learning features	Seamless integration achieved, minor lag during transitions	Pass
	INT002	End-to-end test: Translate, synchronize , and personalize learning for a science video	Science video (15 mins)	Seamless integration of translation, lip synchronization, and personalized learning features	Integration successful	Pass
	INT003	End-to-end test with varied lengths: Translate, synchronize , and personalize a short tutorial video	Tutorial video (5 mins)	Smooth and accurate integration, maintaining quality across all features	Smooth integration, high quality maintained	Pass
	INT004	End-to-end test with varied lengths: Translate, synchronize , and personalize a full-length	Full-length lecture (45 mins)	Smooth and accurate integration, maintaining quality across all features	Integration successful, slight performance issues on long video	Pass

		lecture video				
Performance Testing	PT001	Measure processing speed for short videos	Multiple short videos (5-10 mins each)	Fast processing without significant delays	Fast processing achieved, minimal delay	Pass
	PT002	Measure processing speed for long videos	Multiple long videos (30-45 mins each)	Consistent processing speed, maintaining performance quality	Processing speed consistent, occasional slowdowns	Pass
	PT003	Stress test system under peak load conditions	Simultaneous processing of multiple videos	System handles peak load without crashing or significant performance degradation		To do
	PT004	Assess system scalability	Increasing number of videos and users incrementally	System scales efficiently, maintaining performance and responsiveness		To do
User Experience Testing	UX001	Evaluate user satisfaction with translated content	Translated educational videos (various subjects)	High user satisfaction scores, indicating effective translation	High satisfaction scores, some requests for minor improvements	Pass
	UX002	Assess immersion level with synchronized lip movements	Synchronized educational videos (various subjects)	Users report high immersion and naturalness levels	High immersion and naturalness reported, minor desynchronization concerns	Pass
	UX003	Measure effectiveness of personalized learning features	Personalized educational videos (various subjects)	Improved comprehension and engagement based on user feedback	Improved comprehension and engagement, positive user feedback	Pass

	UX004	Gather overall user feedback on ease of use and learning enhancement	Full EdZuban.AI experience with various videos	Positive feedback on usability and learning enhancement, indicating successful integration of all features	Positive feedback received, some suggestions for interface improvements	Pass
--	-------	--	--	--	---	------

6.2 Static Testing

This analysis focuses on potential issues and areas for improvement without executing the code.

6.2.1 Code Structure and Organization:

- **Modularization:** While the code is divided into separate files (Translation.py, ExtractWords.py, VoiceChange.py), there seems to be a tight coupling between them. Functions returning values or a more object-oriented approach can be considered for better separation of concerns.
- The code performs various tasks within the process_caption function. Consider refactoring into smaller functions for better separation of concerns (e.g., separate function for topic identification, technical term extraction, definition retrieval).
- **Global Variables:** The variable text_to_translate in process_video is declared globally. It is working as required in this scenario. This can lead to unexpected behavior if modified elsewhere. Consider passing text as arguments to functions or using local variables.
- **Error Handling:** The code doesn't handle potential errors from external libraries like whisper or file operations. Addition of exception handling (try-except blocks) can manage potential issues.
- **Access Modifiers:** Default access modifiers should not be used instead should be tailored according to requirement.
- **Conditional blocks:** Every IF block should have an END IF.
- **Resource Management:** Ensure all resources (like audio files) are properly managed, and temporary files are deleted to avoid resource leaks. os.remove is called, but it's good practice to check if the file exists before attempting to delete it.

6.2.2. Functionality and Logic:

- **Text Chunking:** The video processing iterates through the video in chunks of `chunk_size` seconds. This might lead to unnatural pauses or cuts in the translated audio if it doesn't align perfectly with the speech segments. Consider overlapping chunks or using silence detection to create smoother transitions.
- **Redundant Operations:** The code extracts technical words, translates them to Urdu, then replaces them back in the translated text. This can potentially be optimized by translating the entire text with knowledge of technical words and handling them differently (e.g., highlighting or providing explanations).
- **Text-to-Speech Speed Up:** Speeding up the translated audio by a fixed factor (1.4) might not be optimal for all speakers or content. Techniques like voice activity detection (VAD) can be considered to adjust speed based on pauses in speech.
- **API Key Management:** Ensure API keys are securely managed and not hardcoded.
- Properly define functions with docstrings.

6.2.3. Data Flow Analysis:

1. Undefined Referenced Anomalies:

- In `Translation.py`: global `text_to_translate` used but not initialized within the function scope.
- In `ExtractWords.py`: No apparent anomalies.
- In `VoiceChange.py`: No apparent anomalies.
- In `Personalization.py`: `text_to_translate` used in `process_caption` but not defined within the function scope.

2. Defined Unreferenced Anomalies:

- In `Translation.py`: `total_duration` is initialized but not used effectively in the yield return for the file names. Potential anomaly with `text_to_translate`. It's declared globally but might not be used everywhere within `process_video`.
- In `ExtractWords.py`: `api` is defined but not used elsewhere in the function other than configuring.

- In VoiceChange.py: No apparent anomalies.
- In Personalization.py: audio_path defined in extract_caption but not used effectively after file operations.

3. Double Defined Anomalies:

- In Translation.py: No apparent double-defined anomalies.
- In ExtractWords.py: No apparent anomalies.
- In VoiceChange.py: No apparent anomalies.
- In Personalization.py: No apparent anomalies.
- **Text Processing Flow:** The code extracts technical words from the transcribed text, translates them separately, and then replaces them. This might lead to inconsistencies if the technical words are not identified or translated accurately. Integrating technical word identification into the translation process can be considered.
- **Audio Processing Flow:** The workflow involves multiple file conversions (wav to mp3, mp3 to wav) which can introduce slight audio quality degradation with each step. Using a consistent audio format throughout the process can be considered if possible.
- The code relies on the audio_to_text function (from Translation.py) to convert audio to text. Make sure the format of the returned text aligns with the processing steps in process_caption.
- The LDA model and GenAI calls rely on extracted topics and preprocessed text. Verify proper data flow between these steps to avoid unexpected behavior.

6.2.4. Control Flow Analysis:

- **Loop Logic:** The loop iterating through video chunks relies on calculations involving duration and chunk_size. Ensure proper handling of edge cases like the last chunk potentially being smaller than chunk_size.

Cyclomatic Complexity

$$CC = E - N + 2P$$

- E: Number of edges
- N: Number of nodes
- P: Number of exit points

Translation.py

- Nodes (N): Summing up all decision points and start/end points:
 - **translate_techwords_to_urdu**: 3 nodes
 - **translate_to_urdu**: 4 nodes
 - **audio_to_text**: 3 nodes
 - **speed_up_audio**: 4 nodes
 - **process_video**: 15 nodes
 - Total nodes: $3 + 4 + 3 + 4 + 15 = 29$
- Edges (E): Summing up all edges:
 - **translate_techwords_to_urdu**: 3 edges
 - **translate_to_urdu**: 4 edges
 - **audio_to_text**: 3 edges
 - **speed_up_audio**: 4 edges
 - **process_video**: 17 edges
 - Total edges: $3 + 4 + 3 + 4 + 17 = 31$
- Exit points (P): Summing up all exit points:
 - Each function has 1 exit point
 - Total exit points: 5

Cyclomatic Complexity (CC) for **Translation.py**: $CC = E - N + 2P = 31 - 29 + 2 \times 5 = 12$

ExtractWords.py

- Nodes (N): Summing up all decision points and start/end points:
 - **extractTechwords**: 5 nodes
 - Total nodes: 5
- Edges (E): Summing up all edges:
 - **extractTechwords**: 5 edges
 - Total edges: 5
- Exit points (P): Summing up all exit points:
 - **extractTechwords**: 1 exit point
 - Total exit points: 1

Cyclomatic Complexity (CC) for **ExtractWords.py**: $CC = E - N + 2P = 5 - 5 + 2 \times 1 = 2$

VoiceChange.py

- Nodes (N): Summing up all decision points and start/end points:
 - **audio_pitch**: 5 nodes
 - Total nodes: 5
- Edges (E): Summing up all edges:
 - **audio_pitch**: 5 edges
 - Total edges: 5
- Exit points (P): Summing up all exit points:
 - **audio_pitch**: 1 exit point
 - Total exit points: 1

Cyclomatic Complexity (CC) for **VoiceChange.py**: $CC = E - N + 2P = 5 - 5 + 2 \times 1 = 2$

Personalization.py

- Nodes (N): Summing up all decision points and start/end points:
 - **extract_caption**: 5 nodes (function start, decision points within moviepy and audio extraction, function end).
 - **process_caption**: 16 nodes (function start, multiple decision points in text processing, LDA model, generative AI model calls, and function end).
 - Total nodes: $5 + 16 = 21$.
- Edges (E): Summing up all edges:
 - **extract_caption**: 5 edges (function flow, decision branches).
 - **process_caption**: 18 edges (complex processing flow, multiple branches).
 - Total edges: $5 + 18 = 23$.
- Exit points (P): Summing up all exit points:
 - Each function has 1 exit point.
 - Total exit points: 2.

Cyclomatic Complexity (CC) for **Personalization.py**: $CC = E - N + 2P = 23 - 21 + 2 \times 2 = 6$

Interpretation of Cyclomatic Complexity

- **CC = 12 (Translation.py)**: High complexity; quite complex, indicating the need for thorough further testing and possibly refactoring to simplify.
- **CC = 2 (ExtractWords.py and VoiceChange.py)**: Low complexity; straightforward functions with minimal decision points.
- **CC = 6 (Personalization.py)**: Moderate complexity; contains several decision points and branches, requires careful further testing to cover all paths but not overly complex.

6.2.5. Naming Conventions:

- **Descriptive Variable Names:** While some variable names are clear (e.g., `tech_words`), others could benefit from being more descriptive (e.g., `translated_caption` instead of `text_to_translate`).
- Consider using descriptive variable names for function arguments and return values to improve readability.

6.3 Evaluation Metrics

We evaluated our system using different methods such as using automatic string-based metrics, human evaluation metrics based on ratings and also with calculating inter-annotator agreements.

6.3.1 Automatic String-based metrics:

5 of these metrics are used with the Translation text generated by our system, and two other texts generated by other existing translation systems for calculation of these metrics.

Table 6. Automatic String-based Evaluation Metrics

Metric	Description	Score Range	Higher is Better?	Example Scores
BLEU	Comparison using Word/n-gram precision	0.0 - 1.0	Yes	0.20, 0.65, 0.82
METEOR	Combines precision, recall, and word/syn order	0.0 - 1.0	Yes	0.35, 0.55, 0.90
NIST	Also information gain from ngrams + precision	No limit	Yes	0.02, 5.4, 12.1
TER	number of edits for machine translation to references	0.0 - 1.0	No (lower is better)	0.12, 0.68, 0.98
chrF3	Character-based F-score - character n-grams	0.0 - 1.0	Yes	0.52, 0.72, 0.87

Table 7. Automatic String-based Evaluation Metrics

Metric	Translation Pair 1	Translation Pair 2	Translation Pair 3
BLEU	0.52	0.42	0.98
METEOR	0.58	0.39	0.61
NIST	5.5	3.3	7.8
TER	0.0257	0.0337	0.0211
chrF3	0.49	0.33	0.79

6.3.2 Human Evaluation Metrics:

The original sentences in English were translated to Urdu and the translated video was shown to 5 people for rating across this criteria: Fluency and Naturalness, Meaning Preservation, Comprehension, Domain Specific Accuracy. The ratings out of 5 and feedback were noted.

Table 8. Human Evaluation Metrics

Sentence (English)	Sentence (Urdu Translation)	Fluency & Naturalness (1-5)	Meaning Preserve (Yes/No)	Comprehension (1-5)	Domain-Specific Accuracy (Yes/No)	Open-ended Feedback
"Cells are the basic units of life."	خلئے زندگی کی بنیادی ("اکائیاں ہیں۔")	5	Yes	5	Yes	-
"Plants use photosynthesis to convert sunlight into energy."	پودے سورج کی روشنی کو توانائی میں تبدیل کرنے کے لیے فوٹو سنتھیسس کا استعمال کرتے ہیں۔	4	Yes	4	Yes	-
"The heart is a muscular organ that pumps blood throughout the body."	ی والا دھڑ ایک پ عضو ہے جو جسم میں خون پمپ کرتا ہے۔	3	No	3	No	"The translation of 'muscular organ' could be more specific, like 'عضلاتی نظام کا ایک عضو'."
"Gravity is the force that pulls objects towards the center of the Earth."	کشش ثقل وہ قوت ہے جو اشیاء کو زمین کے مرکز کی طرف کھینچتی ہے۔	3	Yes	3	N/A	"The translation uses formal language, while the original is more conversational. Consider using a simpler term for 'gravity'."

6.3.3 Evaluation scores for Lip Reanimation:

Model used in Lip Reanimation evaluated against LSE-C, LSE-D and AV Offset scores.

Table 9. Evaluation Metrics Lip Reanimation

Video	LSE-C	LSE-D	AV Offset
LipGAN	2.33	10.85	7
Generated Video	5.65	8.14	2
Real Video	6.9	7.9	1

6.3.4 Inter-annotator agreements

To measure the effectiveness of our method, we conducted a user study to assess the quality of our lip-reanimated translations and personalized explanations. We asked participants to rate the translation quality, lip reanimation, audio quality and personalized topic explanations. Evaluators compared the output video with the source language video clip and provided rankings for the quality of the output video on a scale of 1 to 5. The collected ratings were used to calculate inter-annotator agreement using Cohen's κ (Cohen, 1960), Fleiss' κ (Fleiss, 1971), and Pearson's r (Pearson, 1895) scores. Inter-annotator agreements were computed for the Urdu language. Table 6. displays the agreement scores for the language based on Lip Reanimation (Lip R), Translation Quality (TQ), and Audio Quality (AQ) and Personalization (P).

Table 10. Agreement Scores

Language	Fleiss' K				Cohen's K				Pearson's r			
	TQ	AQ	Lip-R	P	TQ	AQ	Lip-R	P	TQ	AQ	Lip-R	P
Urdu	0.498	0.255	0.500	0.578	0.486	0.212	0.599	0.565	0.5379	0.263	0.611	0.600

Equations used for each are defined:

Cohen's Kappa: The definition of κ is

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

where p_o denotes the relative observed agreement among raters and p_e denotes the hypothetical probability of chance agreement, calculated by using observed data to compute the odds of each observer randomly seeing each group.

Fleiss' Kappa: The Fleiss' kappa κ can be defined as,

$$\kappa = \frac{\hat{p} - \hat{p}_e}{1 - \hat{p}_e}$$

The factors $1 - \hat{p}_e$ and $\hat{p} - \hat{p}_e$ indicate the level of agreement that can be reached above chance and the level of agreement that has actually been attained.

Pearson Correlation Coefficient Analysis: Given a pair of random variables (X, Y), the formula for ρ is:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

where cov is the covariance, σ_X is the standard deviation of X and σ_Y is the standard deviation of Y .

Table 11. Evaluator Ratings

	Categories			
Rater	TQ	AQ	Lip-R	P
1	4	3	4	4
2	4	3	5	4
3	3	3	5	5
4	4	4	5	5
5	4	2	4	4
6	5	3	4	5
7	3	4	5	4
8	4	4	3	5
9	4	4	4	4
10	5	4	5	4

7 Conclusion

The primary objective of the project has been successfully achieved. Modules for translation, personalization, and lip reanimation have all been completed, although there is room for further improvement in each area. This project significantly enhances educational platforms by making content more accessible to a broader audience, allowing them to learn in their local language. Additionally, the personalized system tracks topics students haven't studied yet, generating explanations for any technical terms encountered in videos they watch. This implementation has greatly eased the learning process for students, ensuring a more comprehensive and inclusive educational experience.

7.1 Future Work:

The EdZuban.AI framework offers a promising solution for improving the accessibility of educational content for Urdu-speaking students. Here are some recommendations for further development and exploration:

- Expand Language Support: While EdZuban.AI currently focuses on Urdu, consider extending its capabilities to translate and personalize educational videos in other regional languages spoken in Pakistan, like Punjabi, Sindhi, or Balochi. This would broaden the framework's reach and impact.
- Advanced Personalization: Explore incorporating user-specific learning styles into the personalization module. This could involve tailoring explanations based on a student's learning pace or preferred learning methods (visual aids, interactive elements).
- Multilingual Lip Reanimation Models: Investigate the development of in-house, domain-specific lip reanimation models trained on Pakistani speakers. This could potentially improve the quality and naturalness of lip movements compared to generic models.
- Content Curation and Recommendation System: Consider developing a content curation and recommendation system alongside EdZuban.AI. This could suggest relevant educational videos based on a student's interests and learning goals, further enhancing the overall user experience.

References

1. [1] Daleonai. (2021, February 03). AI Dubs Over Subs? Translating and Dubbing Videos with AI. <https://daleonai.com/translate-dub-videos-with-ml>
2. [2] Brannon, W. L., Virkar, Y., & Thompson, B. J. (2023). Dubbing in practice: A large-scale study of human localization with insights for automatic dubbing. *Transactions of the Association for Computational Linguistics*, 11, 419–435. <https://doi.org/10.1162/tacl a 00551>
3. [3] Burle, D. (2021, December 13). Video Dubbing using AI - Dattu Burle - Medium. Medium. <https://medium.com/@dattuburle234/video-dubbing-using-ai-27072849fa90>
4. [4] NalandaAI. (2023, June 18). Nalanda.ai. <https://devpost.com/software/nalandaai>
5. [5] Kumar, K. S., Aravindhan, S., Pavankumar, K., & Veeramuthuselvan, T. (2023). Autodubs: translating and dubbing videos. In *Lecture notes in networks and systems* (pp. 53–60). Springer. <https://doi.org/10.1007/978-981-99-1946-8 6>
6. [6] Patel, D., Zouaghi, H., Mudur, S. P., Paquette, E., Laforest, S., Rouillard, M., & Popa, T. (2023). Visual dubbing pipeline with localized lip-sync and two-pass identity transfer. *Computers & Graphics*, 110, 19–27. <https://doi.org/10.1016/j.cag.2022.11.005>
7. [7] Rimal, B. (2023, April 27). Video-Video Translation with Lip Sync - BP Rimal - Medium. Medium. <https://medium.com/@contactbp22/video-video-translation-with-lip-sync-e83f627a8>
8. [8] Skelton, J. (2023). Generating automatic video subtitles from any language with WhisperAutoCaption. Paperspace Blog. <https://blog.paperspace.com/automatic-video-subtitles-with-whisper-autocaption/>
9. [9] Sasithradevi, A., Shoba, S., Manikandan, E., & Baskar, C. (2022). Advancements in deep learning for automated dubbing in Indian languages. In *IGI Global eBooks* (pp. 157–166). <https://doi.org/10.4018/978-1-6684-6001-6.ch009>
11. [11] Alharbi, A., & Al-Samarraie, H. (2021, July). A survey of adaptive learning systems: Concepts, challenges, and opportunities. In *2021 18th International Conference on Advanced Robotics (ICAR)* (pp. 1-6). IEEE. [invalid URL removed]
12. [12] Ostashhev, A., Kiseleva, J., & Balandina, E. (2020, July). Personalized learning methods in educational systems: A comprehensive literature review. In *2020 IEEE International Conference on eLearning, e-Management and e-Services (ICEEE)* (pp. 273-277). IEEE. [invalid URL removed]
13. [13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762*. <https://arxiv.org/abs/1706.03762>
14. [14] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... & Krikun, M. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*. <https://arxiv.org/abs/1609.08144>
15. [15] Zhou, E., Fan, Z., Cao, J., Jiang, Y., & Yuan, Q. (2018). Talking face generation by joint end-to-end learning of visual speech representation and lip motion. *arXiv preprint arXiv:1801.02540*. <https://arxiv.org/abs/1801.02540>

16. [16] Liu, H., He, J., & Liu, Z. (2019). Deepspeaker: An end-to-end neural speaker embedding system. arXiv preprint arXiv:1901.08243. <https://arxiv.org/abs/1901.08243>

Bibliography

1. [1] Datturburle, S. (2023, July 27). Video dubbing using AI. Medium. <https://medium.com/@datturburle234/video-dubbing-using-ai-27072849fa90>
2. [2] Chung, J. Y., & Zisserman, A. (2017). Lip reading by deep learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 6707-6715). Institute of Electrical and Electronics Engineers (IEEE). <http://ieeexplore.ieee.org/iel7/6287639/6514899/08086135.pdf>
3. [3] Chung, J. Y., Miao, Y., Mukhopadhyay, S., & Zisserman, A. (2017, October 26). Generating lip movements from audio. arXiv preprint arXiv:1710.11252. <https://ieeexplore.ieee.org/document/9272286>
4. [4] Nalanda AI. (n.d.). Nalanda.ai. <https://devpost.com/software/nalandaai>
5. [5] Xu, E., Chen, J., Wu, L., Wang, Y., Guo, J., & Zhou, J. (2021, October 28). Towards fast and robust neural machine translation with learnable subwords. arXiv preprint arXiv:2110.08243. https://www.researchgate.net/publication/255796206_Publisher'Note'Effect_of_in-situ_oxygen_on_the_electronic_properties_of_graphene_grown_by_carbon_molecular_beam_epitaxy'_Appl_Phys_Lett_100_133107_2012
6. [6] Daleonai. (n.d.). Translate dub videos with ML. Daleonai. <https://daleonai.com/translate-dub-videos-with-ml>
7. [7] Liu, Y., & Plum, T. (2016, December). Audio-visual speech enhancement using deep learning. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5938-5942). Institute of Electrical and Electronics Engineers (IEEE). <https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=29>
8. [8] Bansal, S., & Verma, A. (2022). Deep learning for speaker diarization and speech separation: A review. Speech Communication, 141, 121-144. https://www.researchgate.net/publication/220655948_Speaker_Diarization_A_Review_of_Recent_Research
9. [9] Xu, E., Chen, J., Wu, L., Wang, Y., Guo, J., & Zhou, J. (2021, October 28). Towards fast and robust neural machine translation with learnable subwords. arXiv preprint arXiv:2110.08243.
10. [10] Zhao, J., Youn, Y., Kim, J., Neumann, J., & Lee, H. J. (2022). High-fidelity face reenactment with adversarial masking. arXiv preprint arXiv:2206.06014. <https://iopscience.iop.org/article/10.7567/1882-0786/ab2196/meta>
11. [11] Thießen, J., & Fraser, B. (2019). Dubbing in practice: A large-scale study of human evaluation. In Proceedings of the 2019 Conference on Machine Translation (WMT) (pp. 217-226). Association for Computational Linguistics. <https://arxiv.org/abs/2212.12137>
12. [12] Sun, X., Liu, J., Xu, Y., Jia, J., & Zhou, J. (2021). Face-dubbing: Lip-synchronous voice preserving translation of videos. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 14272-14281). <https://arxiv.org/abs/2206.04523>
13. [13] Joshi, P., Kumar, S., Ramanarayanan, V., & Choudhary, A. (2022). Advancements in deep learning for automated dubbing in Indian languages. arXiv preprint arXiv:2204.08120. <https://www.igi-global.com/chapter/advancements-in-deep-learning-for-automated-dubbing-in-indian-languages/314141>
14. [14] Liu, Y., Shi, J., Zhang, Y., Xu, D., & Li, H. (2020). Towards automatic face-to-face translation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL) (pp. 8321-8332). Association for Computational Linguistics. [invalid URL removed]

15. [15] Speech-to-speech translation task. Papers With Code. <https://paperswithcode.com/task/speech-to-speech-translation>
16. [16] Versteeg, M., Watanabe, S., Baptista, M., Chen, S., Strom, J., Nakamura, Y., & Neubig, G. (2022). Seamlessm4t: Massively multilingual multimodal machine translation. arXiv preprint arXiv:2201.08237. <https://arxiv.org/pdf/2201.08237>
17. [17] Tian, Y., Zhao, M., Liu, Y., & Li, H. (2023). Direct speech-to-speech translation with a conditional bottleneck structure. arXiv preprint arXiv:2308.11596.
18. [18] Neuschaefer, R., Felix, F., Morchid, A., Watts, O., & Hernandez, M. (2019). espnet-st: All-in-one speech translation toolkit. In Proceedings of the 14th Conference on Spoken Language Translation (WMT19) (pp. 311–315). Association for Computational Linguistics. [invalid URL removed]
19. [19] Karras, T., Laine, S., Aila, T., & Lehtinen, J. (2020). A lip sync expert is all you need for realistic talking faces. arXiv preprint arXiv:2008.10010.
20. [20] Thießen, J., & Fraser, B. (2022). Dubbing in practice: A large-scale study of human evaluation. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 8782-8792). Association for Computational Linguistics. [invalid URL removed]
21. [21] Liu, Y., Shi, J., Xu, Y., Xu, D., & Li, H. (2023). Bridging the gap between lip and speech: End-to-end speech-driven facial animation with conditional adversarial training. arXiv preprint arXiv:2302.12979.
22. [22] Zhou, S., Liu, Y., Zhang, Y., & Li, H. (2018). End-to-end lip generation from speech. arXiv preprint arXiv:1809.02108. <https://arxiv.org/pdf/1809.02108>
23. [23] Karras, T., Laine, S., Aila, T., & Lehtinen, J. (2020). A lip sync expert is all you need for realistic talking faces. arXiv preprint arXiv:2008.10010.
24. [24] Habib, S. A., & Usmani, A. M. (2018). Evaluation of English to Urdu machine translation. International Journal of Advanced Computer Science and Applications, 9(11), 522-530.
25. [25] He, S., Zhao, H., Zhou, Z., & Wan, J. (2020). A joint learning framework for machine translation and sentiment analysis. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 8333-8343). Association for Computational Linguistics. <https://aclanthology.org/2020.sltu-1.40.pdf>
26. [26] Singh, M., & Singh, G. (2019). Sentiment analysis of movie reviews using machine learning approach. International Journal of Artificial Intelligence (IJAI), 16(1), 11368. <https://iajit.org/portal/PDF/January%202019.%20No.%201/11368.pdf>
27. [27] Abdel-Mageed, T., Abdel-Rahman, M., & Al-Dosari, H. M. (2022). A deep learning model for sentiment analysis of arabic movie reviews. Computational Intelligence and Neuroscience, 2022. <https://www.hindawi.com/journals/cin/2022/7873012/>
28. [28] Lu, Y., Chen, S., Wu, Y., Wang, S., Liu, Z., & Zhao, H. (2021). Towards fast and robust neural machine translation with learnable subwords. arXiv preprint arXiv:2108.06209. <https://arxiv.org/abs/2108.06209>