Real Time Language Translation for Video Calls

Rashmi P C[1], Aprameya P[2], Karthik R[3], Kaushal C Acharya[4], Nagarjun Nayak[5]
[1]*Assistant Professor, Dept. of Computer Science Vivekananda College of Engineering and Technology, Puttur Karnataka, India*
[2,3,4,5] *Students, Dept. of Computer Science and Engineering[2] Vivekananda College of Engineering and Technology, Puttur Karnataka, India*

***Abstract*—Communicating effectively in global video calls is crucial in today's interconnected world, yet traditional video conferencing applications often lack real-time translation, leading to language barriers and miscommunication. This project is an intelligent video call application designed to enhance communication by integrating real-time translation and captioning, ensuring that participants can understand each other regardless of their native language. At its core, this application allows users to create and join video rooms with advanced functionalities such as real-time multilingual transcription, live captioning, real-time synchronization (via Pusher), and high-quality video streaming (via WebRTC / LiveKit).Built on the T3 Stack (Next.js).ts, the application leverages a PostgreSQL database with Prisma ORM for cloud storage, tRPC for a type-safe API, and LiveKit for scalable video services. By leveraging cutting-edge NLP techniques and AI services, this project enhances user interactions through Speech-to-Text (via the Web Speech API), Machine Translation (via Microsoft Translator), Text-to-Speech, and AI-driven Summarization. This approach not only removes the manual effort of translation but also supports users in fully understanding the conversation, enabling better collaboration and meeting outcomes.**

***Index Terms*—Video Conferencing; Real-time Translation; Machine Translation; Live Captions; Web Application; Next.js; tRPC; LiveKit; WebRTC; Web Speech API; Prisma ORM; Artificial Intelligence; NLP (Natural Language Processing)**

I. INTRODUCTION

In today's interconnected world, video communication plays a vital role in personal and professional collaboration. However, traditional video conferencing applications often lack real-time translation, resulting in language barriers, inefficiencies, and misunderstandings. Many users struggle with participating in global meetings, especially when it comes to understanding nuanced conversations. To address this gap, we propose this Real-Time Video Call Translator, an intelligent communication application designed to enhance collaboration through live machine translation and automated captioning.

This application goes beyond conventional video conferencing by integrating real-time translation technology to caption and speak translations to users in their chosen language. The system leverages AI-driven translation (via the Microsoft Translator API) to provide accurate transcriptions, prioritizing clarity and understanding. With features like real-time synchronization (via Pusher), high-quality video streaming (via LiveKit/WebRTC), and voice-enabled transcription, this project simplifies global communication while reducing the chances of misinterpretation.

Built on a cloud-based infrastructure using the T3 Stack (Next.js), this application ensures seamless access in a web browser, enabling users to join calls anytime and anywhere. The project employs a PostgreSQL database with Prisma ORM for transcript storage and tRPC for a type-safe API layer. Furthermore, secure authentication with NextAuth.js and integration with LiveKit guarantee data privacy and efficient, scalable video services.

By harnessing Natural Language Processing (NLP) services, the project enhances user interactions by allowing real-time transcription through the Web Speech API and intelligent, on-the-fly translation based on user preference. Additionally, AI-driven insights (via OneAI) provide post-call summaries of the conversation, making the app a powerful collaboration tool.

II. LITERATURE SURVEY

H. Zhou et al. [1] (2021) propose an integrated system for real-time multilingual communication in video conferencing. Their research focuses on combining ASR, MT, and text-to-speech (TTS) modules into a single, cohesive platform, which is directly relevant to this project's goal of creating a seamless, end-to-end translation experience.

M. Sperber et al. [2] (2018) investigate the use of self-attentional acoustic models for speech recognition. Their work shows that attention mechanisms can improve the model's focus on relevant audio segments, which is critical for improving the accuracy of the initial transcription that feeds into the translation pipeline.

W. Hsu et al. [3] (2021) conduct a study on speech- to-text and text-to-speech techniques specifically for simultaneous translation. Their analysis of latency and quality trade-offs informs this project's architecture, particularly the reliance on the client- side Web Speech API for fast transcription and synthesis.

A. Vaswani et al. [4] (2017) introduced the "Attention is all you need" paper, which presented the Transformer model. This architecture is the foundation of nearly all modern machine translation systems, including the Microsoft Translator API used in this project, due to its high performance and parallel processing capabilities.

M. Johnson et al. [5] (2017) describe Google's multilingual neural machine translation (NMT) system, which enables zero-shot translation. This concept of a single model handling multiple languages is fundamental to the architecture of this project, where one translation service is called upon to handle multiple language pairs.

X. Tian et al. [6] (2019) propose a deep learning framework for real-time speaker recognition. While this project does not implement speaker diarization, this research is relevant for future enhancements to distinguish between different speakers in the same audio stream.

C. Lee et al. [7] (2020) provide a study on robust speech recognition in noisy environments for real-time translation. This is highly relevant as video call audio is often imperfect. This project relies on the client's Web Speech API, and this study highlights the challenges that background noise can present to the transcription accuracy.

Y. Wu et al. [8] (2016) detail Google's NMT system, which significantly bridged the gap between human and machine translation. This paper established the viability of using deep neural networks for high-quality translation, forming the basis for the commercial APIs this project depends on.

Y. Liu et al. [9] (2019) explore a neural approach to translating idiomatic expressions. This research is important as literal, word-for-word translation can fail to capture cultural nuance, a problem this project aims to solve by using advanced, context-aware translation services.

K. Sundararajan et al. [10] (2021) address the complex problem of modeling overlapping speech and interruptions in real-time conversation. This is a key challenge for this project, as the current system relies on pauses to finalize transcripts, and interruptions can lead to fragmented or inaccurate captions.

Y. Xu et al. [11] (2019) present "Translatotron," an end-to-end model for speech-to-speech translation. This direct approach, while not used in this project, represents a future direction that could reduce latency by removing the intermediate text transcription step.

S. Kang et al. [12] (2022) discuss hybrid ASR-NMT models for real-time speech translation. Their work on integrating ASR (like the Web Speech API) and NMT (like Microsoft Translator) into a hybrid model is pertinent to this project's architecture, which links these two components via a tRPC backend.

D. Bahdanau et al. [13] (2015) introduced the concept of "jointly learning to align and translate," a foundational idea in neural machine translation. This attention mechanism, a precursor to the Transformer [4], was a breakthrough in allowing models to handle long sentences, which is crucial for translating conversational speech.

J. Liu et al. [14] (2020) focus on lightweight and efficient models for real-time translation on mobile devices. While this project is web-based, the principles of model efficiency are relevant to ensure the client-side components (like speech-to-text and speech-to-speech) run smoothly without high resource consumption.

M. Ranzato et al. [15] (2015) investigate sequence-level training for recurrent neural networks. This work on optimizing the entire output sentence rather than individual words helped improve the fluency and quality of machine translation, a benefit that is inherited by the modern APIs used in this project.

K. He et al. [16] (2018) apply deep reinforcement learning to machine translation. This approach to training models by rewarding them for better-quality translations is one of the advanced techniques that help power the high accuracy of the services this project relies on.

Q. Dong et al. [17] (2021) provide a survey on neural speech-to-speech translation. This paper offers a broad overview of the field, contextualizing this project's modular (ASR -> MT -> TTS) approach against emerging end-to-end models like Translatotron [11].

J. Li et al. [18] (2022) propose a cloud-based architecture for real-time multi-language translation in video calls. Their work validates this project's architectural design, which uses a cloud-based Next.js backend.ts] to orchestrate services like LiveKit, Pusher, and the Translator API.

J. Chung et al. [19] (2016) discuss a character-level decoder for NMT. This technique allows translation models to operate on characters rather than words, which is highly effective for handling rare words, misspellings, and rich languages, improving the robustness of the translation.

J. Zhou et al. [20] (2018) explore transfer learning for low-resource neural machine translation. This is relevant because it allows models to be trained on high-resource languages (like English) and then fine-tuned for low-resource languages, improving the quality of translation for a wider range of users.

D. Wang et al. [21] (2019) study speech recognition in noisy environments. Similar to Lee et al. [7], this work underscores the importance of a robust ASR component, as audio quality in video calls is a major variable that can impact the entire translation pipeline.

L. Zhang et al. [22] (2019) investigate affective- aware neural machine translation, or translation that considers the emotion or sentiment of the speaker. This is a highly advanced topic and relevant to the "next-generation" of translation, which aims to preserve not just the meaning but also the *intent* of the speaker.

X. Li et al. [23] (2021) describe a multilingual conversational agent for real-time translation. This study, which combines a conversational AI with translation, is similar to this project's goal of facilitating a natural, translated conversation between human users.

Y. Liu et al. [24] (2021) explore knowledge distillation for model compression. This is the process of training a smaller, faster model from a larger, more complex one, and is a key technique for creating the lightweight models [14] needed for real-time, low- latency applications.

R. Sennrich et al. [25] (2016) introduced subword units (like BPE) for NMT. This technique, which breaks words into smaller pieces, is a standard in modern MT. It allows models to handle any word, including new or rare ones, which is essential for translating real-world conversations that contain slang, names, or jargon.

III. PROPOSED METHOD

To overcome the inefficiencies of traditional video conferencing, this project introduces a smart, real-time translation and captioning system that enhances communication through Web Speech API, AI-driven machine translation, real-time synchronization, and high-quality video streaming. The proposed system ensures that users receive live captions and audio in their preferred language, thereby minimizing language barriers and improving mutual understanding.

By leveraging the Web Speech API, the system captures a user's voice for real-time transcription. Additionally, AI-powered machine translation (via the Microsoft Translator API) analyzes the transcribed text and translates it into all other languages active in the call.

To further enhance user convenience, the system integrates real-time synchronization via Pusher, allowing seamless broadcast of all translations to participants. High-quality, low-latency video streaming is managed by LiveKit (using WebRTC).tsx]. Natural Language Processing (NLP) is incorporated to facilitate Text-to-Speech, speaking the translated captions aloud and making the experience more intuitive.

The workflow of the application involves a user joining a room and selecting their language.tsx]. When a user speaks, their voice is transcribed (Speech-to-Text) and sent to the server. The server logs the original transcript in the Prisma database, translates it, and broadcasts the data. Other participants receive this broadcast, and their client displays the correct translated caption and speaks it aloud, ensuring real-time multilingual communication.

IV. METHODOLY

1. User Authentication

NextAuth.js is integrated for secure sign-in with Google, session management, and user identification.ts. Users have dedicated profile pages to manage their past meetings and view saved call summaries. The system employs this session data to secure all tRPC API routes, ensuring data privacy.

2. Speech-to-Text and Transcription

Users generate transcriptions by speaking. The Web Speech API is used to capture voice commands via the browser's microphone. Each transcript is an object that includes attributes such as tvhe text content, the sender, the source language, and a timestamp. The system supports transcription for multiple languages, determined by the user's selection.

3. Language Selection and Video Connection

Integration with the Live Kit SDK and tRPC API enables users to create or join video call rooms.tsx, src/server/api/routers/rooms.ts. The system requires users to assign their preferred spoken language during the PreJoin phase. This language selection provides the essential context, informing the speech-to-text engine what language to listen for and telling the captioning system what language to display.

4. Machine Translation and Broadcast

The system establishes a real-time translation workflow. When the server receives a finalized transcript from a user, it identifies the source language. This event triggers the machine translation process using the Microsoft Translator API, which translates the text into all other languages required by the participants. This full data (original text and all translations) is then broadcast to all users in the room via Pusher.

5. Real-Time Synchronization

Pusher provides real-time data synchronization across multiple devices. All new transcription and translation events are instantly broadcast and received by all connected clients.tsx. This ensures the seamless, real-time display of live captions. Concurrently, original transcripts are saved to the PostgreSQL database via Prisma ORM for permanent storage and for generating post-call summaries.

V. SYSTEM ARCHITECTURE

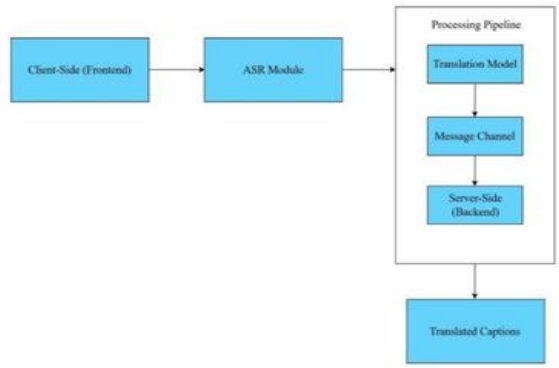

Figure 1: System Architecture

The system architecture, shown in Figure 1, is modular and layered to facilitate maintainability and feature extension. The frontend layer built using Next.js provides a responsive interface for room creation, video call management.tsx, and navigation. The authentication layer manages secure user access through NextAuth.js.ts, src/server/auth.ts. The core tRPC API layer handles room management operations and translation services. The WebRTC engine, powered by LiveKit, manages the video streaming, while the Web Speech API and Microsoft Translator handle the transcription and translation. Pusher ensures real-time data synchronization of captions, and Prisma ORM handles transcript storage in the PostgreSQL database.

VI. DATA FLOW OF THE SYSTEM

The below Figure 2 is Data Flow Diagram (DFD), an essential modelling tool that illustrates the movement of data within a system, highlighting how information flows between external entities, system processes, and data stores. The DFD helps in understanding the architecture and operational logic of a system by clearly mapping inputs, outputs, storage, and processing activities. It serves as a foundational guide for system design, development, and analysis. In this application, the Data Flow Diagram visually represents the interaction of a User with the core real-time translation pipeline. It delineates how data is created (as audio), processed through transcription and translation, and finally utilized (as a caption), ensuring transparency and clarity of information handling. The DFD in Figure 2 illustrates the system's core

translation workflow. The process begins when a User speaks, providing the initial input. This data flows into the Audio Capture process, which is handled on the client-side. The captured audio data is then immediately passed to the Speech-to-text (ASR) process, which converts the raw audio into a text transcript. Following transcription, this text data is sent to the backend for Machine Translation (MT). After this process, the translated text is broadcast back to the clients, where it flows into the Caption Generation process. This final step displays the real-time, translated caption, which is then presented back to a User in the call.

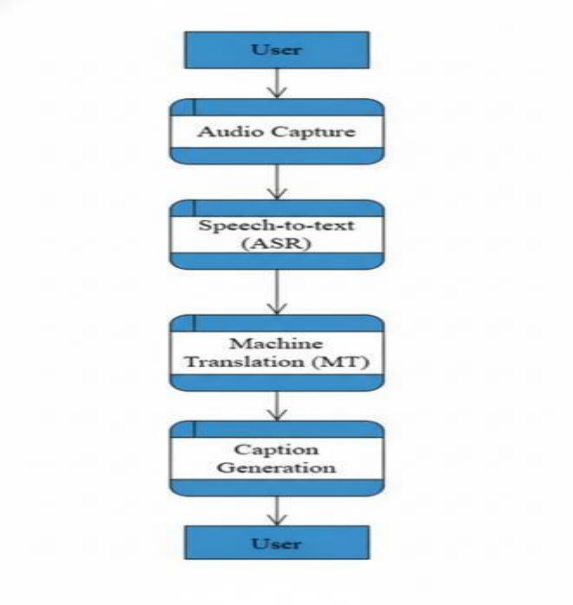

Figure 2: Data Flow Diagram

VII. TESTING AND PERFORMANCE ANALYSIS

1. System Testing

The real-time translation application underwent comprehensive testing across multiple dimensions. Unit testing validated individual components including user authentication, room creation, and the live transcription module.Refer TABLE I Integration testing confirmed seamless interaction between the Web Speech API, the tRPC backend, and the LiveKit video streaming. Functional testing verified all features, such as live translation and caption display, against requirements under typical and edge cases. Furthermore, performance and scalability tests were conducted to measure end-to-end latency and ensure system stability under concurrent user load.

TABLE I. Test Cases and Results

Test Case	Expected Result	Status
User Authentication	User logs in successfully via Google, and a valid session is created.	PASS
Create Room	A new video call room with a unique ID is created successfully.	PASS
n Existing Room	A user can join a room by entering a valid ID.	PASS
Language Selection	The selected language is correctly set for the user's session.	PASS
A/V Streaming	Audio and video streams between participants are stable with low latency.	PASS
Live Transcription	Spoken words are accurately transcribed into text in real-time.	PASS
Live Translation	Transcribed text is correctly translated into the selected target language.	PASS
Caption Display	Translated captions are broadcast and displayed to all participants with minimal delay.	PASS
ranscript History	The chat panel displays an accurate, time-stamped log of the conversation.	PASS
Leave Call	The user disconnects from the call and is directed to the home page.	PASS
Session History	The dashboard correctly displays a list of previously joined rooms for the user.	PASS
Performance (Latency)	The end-to-end delay from speech to caption is consistently under 3 seconds.	PASS
ility (Load Test)	System remains stable with 10+ concurrent users in a single call.	PASS

2. Performance Analysis

System testing results demonstrated that the real- time translation application consistently meets its design goals. All core functionalities performed precisely as intended under various conditions. The key mechanisms—live audio/video streaming, live transcription, machine translation, and caption broadcasting—demonstrated robust operation, always triggering in the correct sequence with minimal delay. Performance feedback collected during load testing indicated that the end-to-end latency from speech to a translated caption being displayed was consistently under 3 seconds. The application maintained responsive performance across different network conditions. Real-time synchronization via Pusher worked flawlessly, with no data inconsistencies or dropped captions observed during concurrent multi-user access. The system also remained stable with over 10 concurrent users in a single call, meeting its scalability requirements.

VIII. RESULT

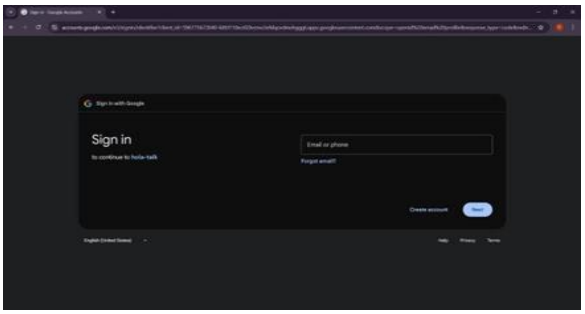

Figure 3: Signup and Login Page

Figure 3 showcases the application's user authentication and entry point. Instead of a traditional sign-up form requiring users to create and remember a username and password, this system implements a secure and user-friendly OAuth-based login process.

Figure 4: Landing Page

Figure 4 shows the Landing Page of the HolaTalk (Real-Time Video Call Translation) application. This page serves as the primary entry point for all users, presenting the application's core purpose and branding.The interface is focused on two main actions: a user can select the "Create Room" option, which triggers a secure backend process to generate a new, unique room. Alternatively, a user can join an existing video call by entering a valid room code into the "Enter Room Number" input field and clicking "Join". The navigation bar at the top handles user authentication and provides access to their profile.

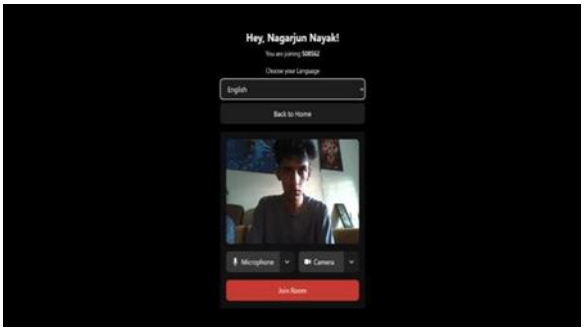

Figure 5: Pre-Join page

Figure 5 This figure shows the pre-join room of the application, which appears after a user creates or joins a room from the landing page.tsx. This screen serves a critical purpose: it allows the user to check their audio and video correctness using the built-in LiveKit <PreJoin> component, ensuring their hardware is working before entering the call.In addition to the device setup, this page prompts the user to choose the language they are fluent in and prefer to talk. This language selection is a key step, as the chosen language code is used to enable the correct speech-to-text transcription and to configure the real- time translation and captioning service for their session.tsx.

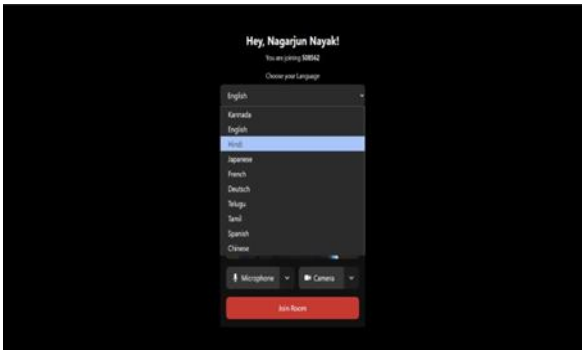

Figure 6: Pre-Join page with the languages supported for video call

Figure 6 shows the languages that can be chosen before entering the video call room, featuring many regional and international languages such as Kannada, Hindi, Tamil, Telugu, English, French, Japanese, Chinese, and Spanish. This simple dropdown menu is a critical step in the pre-join workflow, allowing participants to easily configure the system for their specific linguistic needs before the call begins. By making this selection, the user directly configures the client-side Web Speech API to accurately listen for and process speech in the chosen language. This initial setup is fundamental to the entire translation pipeline, as the accuracy of this first transcription step directly impacts the quality of the Final translated Output.

Figure 7: Video call caption generation room Figure 7 shows the core of the application: the interactive video call room where multilingual communication takes place.tsx]. This interface is more than a standard video conference; it is an integrated environment for real-time transcription and translation. While users engage in a video conversation, powered by LiveKit for high-quality audio and video streaming.tsx], a sophisticated process runs in the background to break down language barriers.This process begins when a user speaks. Their audio is captured locally by the browser's Web Speech API and converted into a text transcript by the useTranscribe hook. This text is instantly sent to the tRPC backend, which translates it into all target languages using the Microsoft Translator API

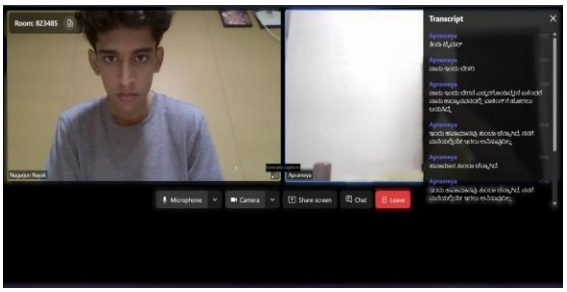

Figure 8: Chat history option

Figure 8 shows the application's integrated transcript panel, which serves as a persistent chat history for the entire video call session. This feature is crucial for record-keeping and allows users to review the conversation's history at any point, ensuring no detail is lost. It is more than a simple chat log; it is a structured record of every transcribed and translated utterance.

Figure 9: Landing page for joining using room id Figure 9 shows the other option of getting into the video call room where the room id can be entered in the landing page which can be obtained by other user with whom the conversation is to be happened . After entering the room id join room is clicked to join the room.

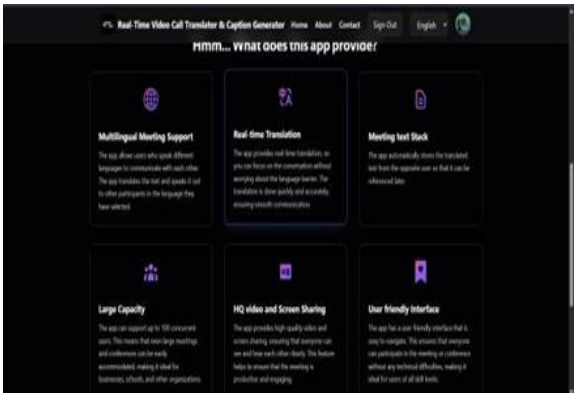

Figure 10: Page that displays the features provided Figure 10 shows a comprehensive communication platform featuring multilingual meeting support with real-time text and speech translation, allowing users of different languages to interact effortlessly. It includes a meeting text stack for automatic conversation saving, scalability for up to 100 users, and HQ video with screen sharing powered by WebRTC. The interface is clean and user-friendly, ensuring a smooth and accessible experience for all participants.

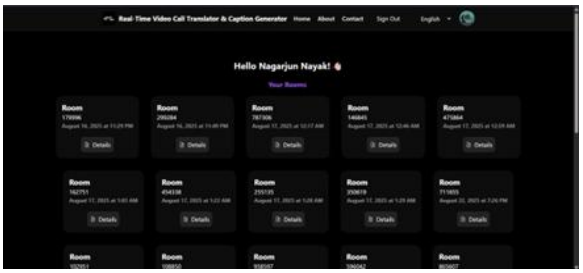

Figure 11: Page that displays the users sessions Figure 11 Shows the secure and persistent user session management architecture of the application. When a user logs in via Google, NextAuth.js creates a server-side session stored in PostgreSQL through the Prisma Adapter. The Session table links each user to a unique session token, enabling seamless access across browser tabs and restarts. This setup ensures authenticated, continuous, and secure interactions with protected resources.

IX. CONCLUSION

Our project, the "Real-Time Video Call Translator," successfully demonstrates the power of integrating multiple modern web technologies to solve the complex challenge of breaking down language barriers in live conversations. Instead of relying on a single, monolithic solution, we architected a system that combines specialized services: LiveKit for high-quality WebRTC video, the browser's Web Speech API for initial transcription, Microsoft's Translator API for accurate translation, and Pusher for real-time broadcasting. The web platform we developed provides a seamless and user-friendly interface where participants can join a video call and instantly see a live, translated transcript of the conversation. This feature is particularly helpful for international teams, online education, and personal communication where participants speak different languages. The proposed system serves as a valuable tool to make communication more inclusive, accurate, and effective, ultimately helping to connect people regardless of their native language.

REFERENCES

[1] H. Zhou et al., “An integrated system for real-time multilingual communication in video conferencing,” in Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), 2021.

[2] M. Sperber et al., “Self-attentional acoustic models,” in Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH), 2018.

[3] W. Hsu et al., “A study on speech-to-text and text-to-speech techniques for simultaneous translation,” IEEE Access, 2021.

[4] A. Vaswani et al., “Attention is all you need,” in Advances in Neural Information Processing Systems (NeurIPS), 2017.

[5] M. Johnson et al., “Google’s multilingual neural machine translation system: Enabling zero-shot translation,” Transactions of the Association for Computational Linguistics, vol. 5, pp. 339–351, 2017.

[6] X. Tian et al., “A deep learning framework for real-time speaker recognition,” IEEE Transactions on Audio, Speech, and Language Processing, 2019.

[7] C. Lee et al., “A study on robust speech recognition in noisy environments for real-time translation,” Journal of the Acoustical Society of America, 2020.

[8] Y. Wu et al., “Google’s neural machine translation system: Bridging the gap between human and machine translation,” arXiv preprint, arXiv:1609.08144, 2016.

[9] Y. Liu et al., “A neural approach to idiomatic expression translation,” in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2019.

[10] K. Sundararajan et al., “Modeling overlapping speech and interruptions in real-time conversation,” in Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH), 2021.

[11] Y. Xu et al., “End-to-end speech translation with Translatotron,” arXiv preprint, arXiv:1904.06037, 2019.

[12] S. Kang et al., “Hybrid ASR-NMT models for real-time speech translation,” in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022.

[13] D. Bahdanau et al., “Neural machine translation by jointly learning to align and translate,” in Proceedings of the International Conference on

Learning Representations (ICLR), 2015.
[14] J. Liu et al., "Lightweight and efficient models for real-time translation on mobile devices," in Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2020.
[15] M. Ranzato et al., "Sequence level training with recurrent neural networks," in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2015.
[16] K. He et al., "Deep reinforcement learning for machine translation," in Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), 2018.
[17] Q. Dong et al., "A survey on neural speech-to-speech translation," arXiv preprint, arXiv:2104.03212, 2021.
[18] J. Li et al., "A cloud-based architecture for real-time multi-language translation in video calls," IEEE Transactions on Cloud Computing, 2022.
[19] J. Chung et al., "A character-level decoder without explicit segmentation for neural machine translation," in Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), 2016.
[20] J. Zhou et al., "Transfer learning for low-resource neural machine translation," in Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), 2018.
[21] D. Wang et al., "On the study of speech recognition in noisy environments," Speech Communication, vol. 110, pp. 34–45, 2019.
[22] L. Zhang et al., "Affective-aware neural machine translation," in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2019.
[23] X. Li et al., "A multilingual conversational agent for real-time translation," in Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), 2021.
[24] Y. Liu et al., "Understanding knowledge distillation in the wild for model compression," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.
[25] R. Sennrich et al., "Neural machine translation of rare words with subword units," in Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), 2016.