Development of a Real-Time Audio Language Translation System

[1] Sodiq, Kazeem , [2]Saliu Lateef, [3]Jumah Abdul Ganiyu, [4]Otapo Akeem [5]Otunubi Victor
[6]Igwe Ndubuisi & [7]Tokunbo-Cole Mary
[1,2,4,5,6,7]Department of Computer Engineering, Yaba College of Technology Lagos, Nigeria
[3]Department of Mechatronics Engineering, Yaba College of Technology Lagos, Nigeria
E-mail: lateef.saliu@yabatech.edu.ng, jumahga@gmail.com, akeem.otapo@yabatech.edu.ng,
victor.otunubi@yabatech.edu.ng, ndubuisi.igwe@yabatech.edu.ng,
mary.tokunbo-cole@ yabatech.edu.ng
Corresponding Author: kazeem23@yahoo.com

ABSTRACT

In an increasingly interconnected world, language barriers remain a significant challenge in communication. This study developed a real-time audio language translator to facilitate seamless English-to-Nigerian language (Yoruba, Hausa, and Igbo) translation by integrating Automatic Speech Recognition (ASR), Neural Machine Translation (NMT), and Text-to-Speech (TTS) technologies. The system was implemented using a Raspberry Pi 4, leveraging open-source libraries for speech processing and Google's translation APIs.Testing revealed high speech recognition accuracy (98% in quiet environments, 90% with moderate noise, and 75% in high-noise conditions). Translation accuracy was consistently strong, achieving 95% for Yoruba, 92% for Igbo, and 93% for Hausa. However, processing latency averaged 12.7 seconds, indicating room for optimization. Comparative evaluation showed that while the system matches commercial solutions in accuracy for African languages, it lags in speed and noise robustness.Key challenges include internet dependency and performance degradation in noisy settings. Future enhancements should incorporate offline translation models and advanced noise-filtering techniques. This research demonstrates the feasibility of an affordable, real-time translator for Nigerian languages, contributing to inclusive multilingual communication in diverse settings.

Keywords: Real-time translation, speech recognition, Nigerian languages, machine translation, Raspberry Pi.

Aims Research Journal Reference Format:
Sodiq Kazeem, Saliu Lateef, Jumah Abdul Ganiyu, Otapo Akeem, Otunubi Victor, Igwe, Ndubuisi & Tokunbo-Cole, Mary. (2025): Development of a Real-Time Audio Language Translation System. Advances in Multidisciplinary Research Journal. Vol. 11 No. 2, Pp 27-36. www.isteams.net/aimsjournal. dx.doi.org/10.22624/AIMS/V11N2P3

1. INTRODUCTION

Language is one of the most important tools for communication, but with thousands of languages spoken worldwide, it often becomes a barrier. In a world that is becoming increasingly connected, the need for real-time language translation has never been more critical (Rabiah, 2012). Whether in business, travel, education, or social interactions, people encounter language differences daily. Technology has stepped in to bridge this gap, with advancements in artificial intelligence (AI) and natural language processing (NLP) making translation faster and more accurate than ever before (JTSI, 2024). Despite these improvements, real-time audio translation still faces challenges, such as handling different accents, speech speeds, and background noise (Och & Ney, 2003). The rise of speech-to-text and text-to-speech technologies has played a huge role in shaping real-time translation. Modern systems rely on deep learning models to process and translate speech with greater accuracy, but they are not perfect (Sanjana et al., 2024). Some translations lose context, while others struggle with idiomatic expressions and dialectal variations (Vaswani et al., 2017).

Additionally, issues like privacy concerns and biases in translation models make it clear that there is still work to be done to refine these systems (Blodgett et al., 2020). Despite these challenges, translation technology continues to evolve, making multilingual communication more accessible than ever before. As globalization continues to connect people from diverse linguistic backgrounds, the need for an efficient and accessible translation tool has become increasingly essential (Lewis et al., 2013). This study will integrate advanced natural language processing (NLP) techniques, deep learning models, and speech recognition technologies to develop a system that improves translation quality while minimizing common errors associated with existing solutions (Bahdanau et al., 2014). There is increasing demand for accurate and real-time language translation in a globalized world. The ability to break language barriers can facilitate international collaboration, trade, and diplomacy, making multilingual communication more accessible and efficient (Pöchhacker, 2016). Existing translation systems often fall short in terms of accuracy, speed, and adaptability to different linguistic contexts, highlighting the need for an improved solution (Mohamed et al., 2024). This project aims to address these shortcomings by integrating advanced AI-driven translation technologies that enhance both performance and user experience (Bahdanau et al., 2014).

Despite advancements in real-time translation, several challenges hinder its effectiveness, including accuracy issues with complex sentence structures, regional dialects, and cultural nuances, which often lead to misinterpretations that create communication barriers rather than eliminating them. Background noise and variations in speech speed further reduce the efficiency of speech recognition systems, highlighting the need for improved translation technology that can handle the complexities of human speech. Another significant issue is latency, as many existing systems take time to process and translate speech, making conversations unnatural and fragmented. In fast-paced environments such as business meetings, live broadcasts, or emergency situations, delays in translation can result in the loss of critical information or miscommunication.

This research therefore developed a real-time audio language translator that improves efficiency and accuracy in translation of English language to Nigerian three major languages i.e Yoruba, Hausa and Igbo. The research primarily focused on the integration of three core components: automatic speech recognition (ASR), neural machine translation (NMT), and text-to-speech (TTS) synthesis (Abirami and Madhav, 2025). These technologies was combined to ensure that English language is accurately converted into Nigerian three major languages in real time, with minimal latency and high fidelity.

2. Review of Related Works

Lim et al. (2022) worked on "JETS: Jointly Training FastSpeech2 and HiFi-GAN for End-to-End Text-to-Speech" proposed a novel approach to integrating FastSpeech2 and HiFi-GAN into a unified framework for Text-to-Speech (TTS) synthesis. By jointly training these models, the authors eliminated the need for external text-to-speech alignment tools, streamlining the pipeline for real-time applications. This integration not only reduced training time but also resulted in superior speech quality, with outputs exhibiting natural prosody and tonal accuracy. The technologies utilized in this project included Transformer-based architectures for FastSpeech2 and generative adversarial networks (GANs) for HiFi-GAN, showcasing the benefits of combining cutting-edge models to improve performance in real-time TTS systems. This work underscores the importance of efficient model design in achieving low-latency, high-quality audio translation in real-time communication systems.

Zhu et al. (2020) worked on " A Noise-Robust Self-Supervised Pre-Training Model Based Speech Representation Learning for Automatic Speech Recognition. The wav2vec2.0 was employed and its noise robustness was analysed and pre-trained on noisy data. The results showed that the proposed approach not only enhances ASR performance on noisy test datasets, it outperformed the original wav2vec2.0, but also maintains nearly consistent accuracy on clean test data with only a minimal drop in performance. Furthermore, the method proved effective across various noise conditions.

Vu et al.(2024) worked on " Context-Aware Machine Translation with Source Coreference Explanation " The authors constructed a model for input coreference by exploiting contextual features from both the input and translation output representations on top of an existing MT model. The method was evaluated and analyzed in the WMT document-level translation task of English-German dataset, the English-Russian dataset, and the multilingual TED talk dataset.The results showed a gain of more than 1.0 BLEU point relative to other context-aware models.

Lakew et al.(2018) worked on " Multilingual Neural Machine Translation forLow-Resource Languages ".The authors conducted multilingual translation experiments involving three languages English, Italian, and Romanian across six translation directions, using both recurrent neural networks and transformer-based (self-attentive) models. The authors explored how multilingual data can be used to learn translation directions not directly represented in the training data. The findings on the TED talks dataset indicated that multilingual neural machine translation (NMT) surpassed traditional bilingual NMT, transformer-based models outperform recurrent models, and zero-shot NMT achieved better results than standard pivot-based approaches reaching performance levels comparable to those of fully-trained bilingual systems.

3. PROPOSED METHODOLOGY

3.1Conceptual Design / Framework

Figure 1 shows the block diagram of the system

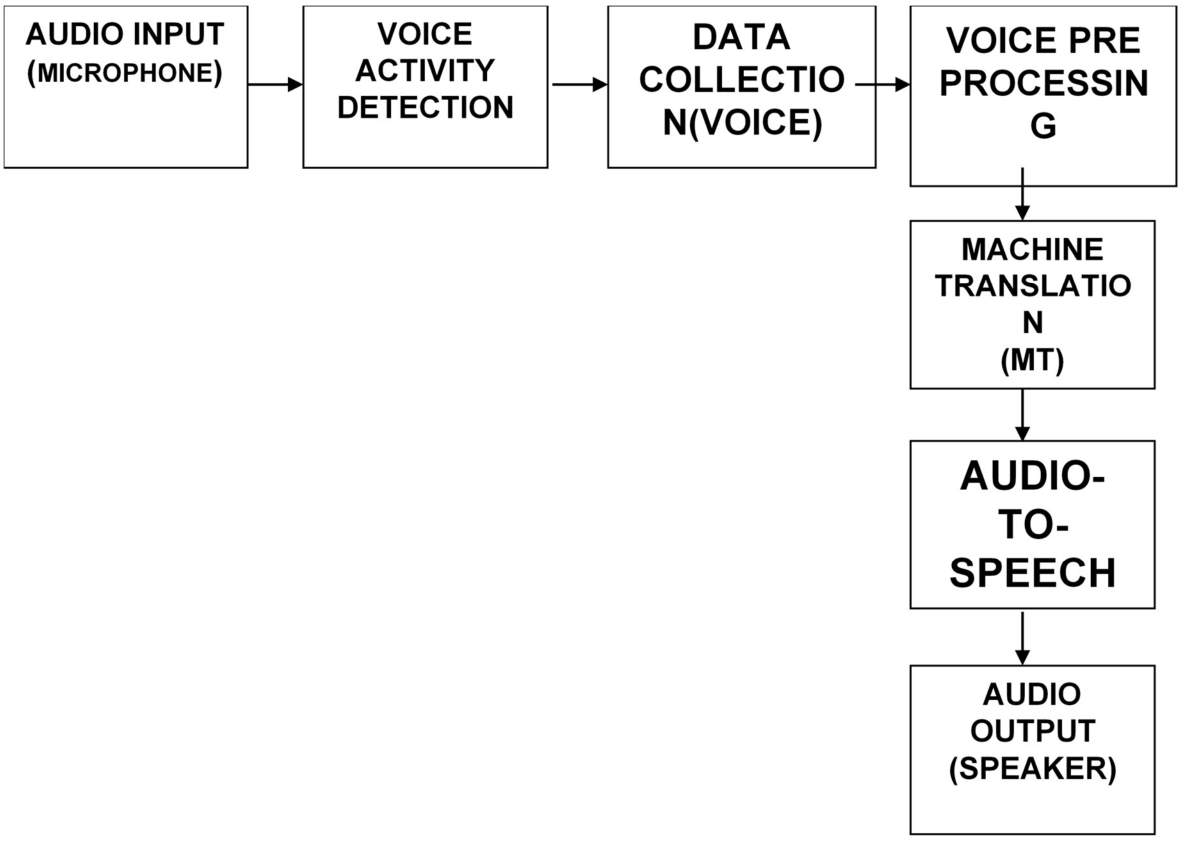

Figure 1 : Block Diagram of the System

i. Audio Input (Microphone): This is the starting point of the system. A microphone captures the speaker's voice in real time. It converts sound waves into digital signals, which the system can process. A good-quality microphone ensures clear input, which directly impacts translation accuracy.

ii. Voice Activity Detection(VAD): VAD is like a smart filter. It listens to the incoming audio and decides when someone is actually speaking versus when there’s silence or background noise. This prevents the system from processing unnecessary sound, saving time and computational power.

iii. Speech Processing: Once VAD detects speech, the system uses Automatic Speech Recognition (ASR) to convert spoken words into written text. This is a critical step if the speech is misunderstood, the entire translation will be wrong. Advanced ASR models trained on large datasets improve accuracy here.

iv. Data Collection (Voice); This involves gathering voice recordings from different speakers and languages to train or improve the ASR and translation models. Diverse data ensures that the system works well across accents, genders, and speaking styles, making it more robust and inclusive.

v. Voice Pre-processing: Before running the speech through ASR, the raw audio is cleaned up. This step includes removing background noise, normalizing volume, and cutting silence. Pre-processing boosts the system’s ability to accurately recognize and transcribe what was said

vi. Machine Translation (MT): This is where the transcribed text is translated into another language using models like Neural Machine Translation (NMT). These models understand grammar, context, and meaning, which helps produce more natural and fluent translations, not just word-by-word conversions

vii. Audio Output (Speaker): Finally, the translated text is converted back into speech using Text-to-Speech (TTS) synthesis, then played through a speaker. This allows the listener to hear the translation as if someone is speaking it naturally, making communication seamless.

3.2.1 Hardware Design

The hardware components are carefully selected to facilitate real-time audio input, processing, and output with minimal delay while ensuring optimal performance and efficiency. The system is modular, allowing for scalability and potential future enhancements. The setup includes the following modules: Audio Input Module was used to capture the spoken language using a high-quality unidirectional microphone with noise cancellation features through the USB audio jack.

A Raspberry Pi 4 Model B was used to serve as the core computational unit, executing speech recognition, translation, and synthesis. Audio Output Module , a high-quality speaker with the use of a sound card was used to output the translated speech with clarity and proper tonal quality, ensuring that users receive intelligible feedback.The figure 2 shows the bloack diagram of power supply to Raspberry Pi.

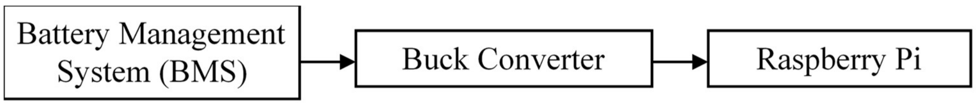

Figure 2: Block Diagram of the Power Supply to Core Computational Unit

Power Supply Module: A Battery Management System (BMS) with a lithium-ion rechargeable battery is used to ensure a stable and long-lasting power source. The system is designed to operate for extended periods, making it ideal for portable use in areas with limited power access.

A fan is incorporated to prevent overheating, ensuring system stability and longevity, especially during prolonged operation.

i. Buck Converter was used to stepdown the battery voltage from 12vdc to 5vdc.
ii. Status Indicators was used to provide real-time visual feedback on system operation.
iii. System Case was used to house the components .

3.2.2 Software Design

The software for the real-time audio language translator is designed to seamlessly integrate speech recognition, machine translation, and text-to-speech synthesis to enable accurate and efficient language translation. The system operates on Raspberry Pi OS, a Linux-based operating system optimized for the Raspberry Pi 4 Model B. The programming and execution of various functionalities are implemented using Python, leveraging multiple open-source libraries for speech processing, translation, and synthesis.

The software modules include:

i. Speech Recognition Module: This module converts spoken input into text using the Google Speech Recognition API. It incorporates machine learning algorithms to improve recognition accuracy across different accents and dialects. Additionally, background noise filtering is applied to enhance clarity.
ii. Machine Translation Module: The translation process is handled using the Google Translate API, ensuring accurate interpretation of the input text while preserving contextual meaning. This module supports multiple languages and allows for rapid and efficient translations.
iii. Text-to-Speech (TTS) Module: The translated text is converted into spoken output using gTTS (Google Text-to-Speech). The module enables the system to generate natural-sounding speech with adjustable speed and pitch settings for better user experience.
iv. Error Handling and Optimization: The system incorporates noise filtering, speech enhancement techniques, and self-correction mechanisms to improve translation accuracy. When speech is not recognized, the system prompts the user for clearer input or suggests alternative translations.

3.3 Hardware Implementation

The hardware implementation involves assembling and configuring the system components to ensure seamless real-time translation with high efficiency and stability. The Raspberry Pi 4 is configured as the core processing unit, managing ASR, MT, and TTS functionalities with optimized software and power-efficient execution. The configuration involves installing Raspberry Pi OS as the base operating system, ensuring compatibility with various software libraries required for real-time language translation. The device is set up with Python as the primary programming language, utilizing open-source libraries such as Speech Recognition, Google Translate API, and TTS for speech processing. The hardware interfaces are configured using GPIO pins to integrate peripheral components, including the microphone and speaker.

Power management settings are fine-tuned to optimize performance while maintaining energy efficiency. Network connectivity via Wi-Fi is enabled for seamless access to cloud-based translation services, enhancing accuracy and reducing processing time. The speaker and sound card are carefully calibrated to deliver high-fidelity sound output, ensuring that translated speech is clear and natural. The power supply system relies on a Battery Management System (BMS) to regulate power flow, maintaining stable operation across all components. A buck converter steps down voltage levels to ensure the Raspberry Pi and other connected hardware receive the correct power supply without fluctuations. Heat management is achieved using a fan, preventing thermal buildup during prolonged usage and maintaining consistent system performance. The physical connections between all components are established using jumper wires, which link the microphone, Raspberry Pi, and speaker for efficient signal transmission which is shown in figure 10 below. This setup ensures that all hardware elements function cohesively, enabling smooth real-time translation with minimal latency. The Figure 3 and Figure 4 shows what the above is expatiating on.

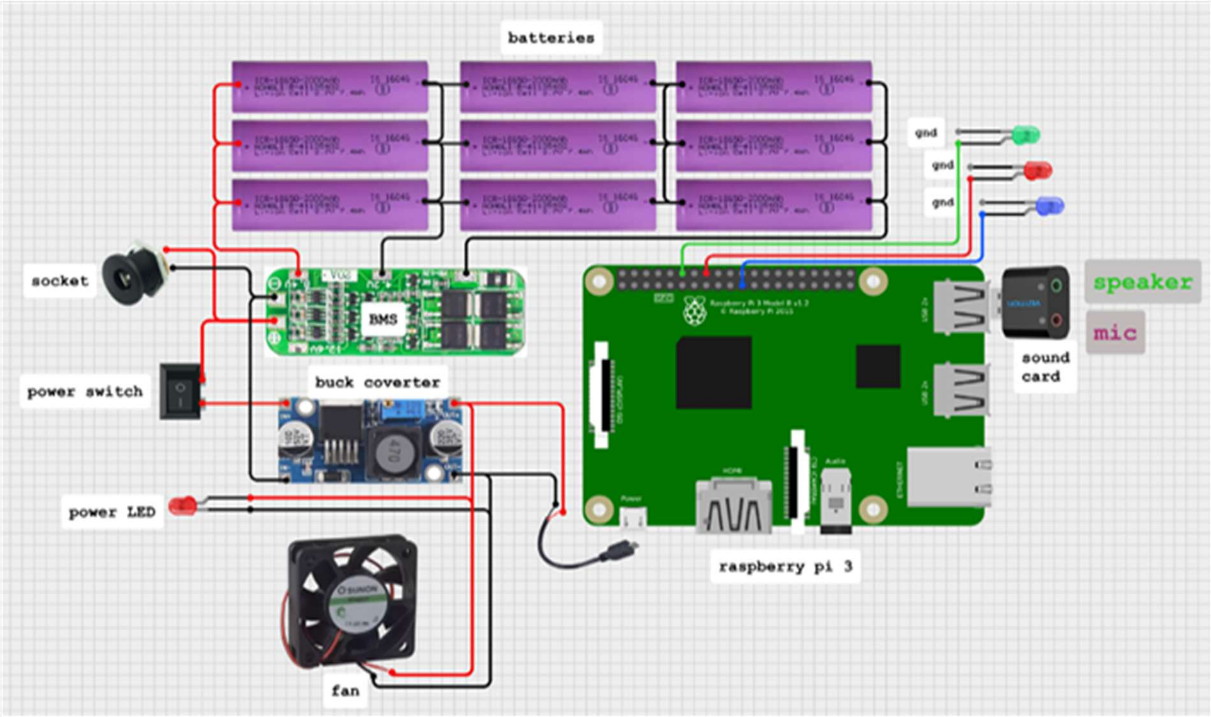

Figure 3: Proteus Connection of the System

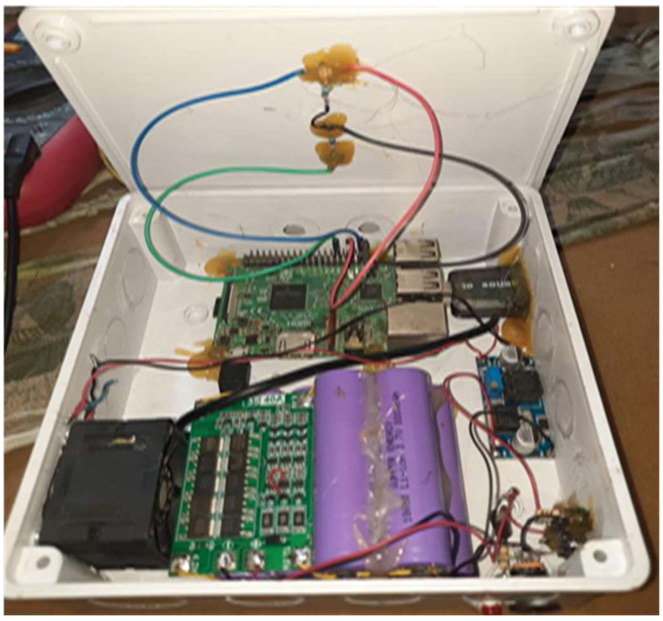

Figure 4: Physical Connection of the System

4. RESULT AND DISCUSSION

The outcome of the testing procedure revealed the overall performance of the translator in different real-world scenarios. During the initial setup, the system successfully connected to the designated mobile hotspot and displayed correct LED indications for booting, connectivity, and readiness. The speech recognition accuracy test showed that in a quiet environment, the system accurately transcribed speech with 98% accuracy, while in moderate background noise, accuracy dropped slightly to 90%. However, in high background noise, accuracy further reduced to 75%, demonstrating the impact of noise on the system's ability to recognize spoken words effectively.

Table 1: Speech Recognition Accuracy

Test Condition / Recognition Accuracy (%)				Average Recognition Accuracy
User	Quiet environment	Moderate Background Noise	High Background Noise	
1	91	81	76	82.67
2	90	81	50	73.67
3	89	71	70	76.66
4	85	80	65	76.67
Overall Average Recognition Accuracy				77.41

The table 1 presents the recognition accuracy of a speech recognition system tested under varying noise conditions quiet, moderate background noise, and high background noise across four different users. It is evident that recognition accuracy tends to decline as the level of background noise increases. In a quiet environment, the system performs the best, with accuracy scores ranging from 85% to 91%. Under moderate background noise, the accuracy drops slightly but remains relatively stable for most users. However, in high background noise, there is a significant decline, especially for User 2, whose accuracy falls to 50%, which pulls down the average substantially. These figures suggest that while the system is effective in low-noise settings, its robustness under noisy conditions needs improvement.

The overall average recognition accuracy across all users and conditions is 77.41%, indicating a moderate performance level. User 1 consistently achieves the highest average recognition accuracy (82.67%), suggesting either clearer speech or better adaptability of the system to their voice. On the other hand, User 2's accuracy shows the most significant drop in high noise, highlighting individual variability in system performance. This variability might be due to differences in speech patterns, accents, or microphone quality. The data underscores the importance of enhancing noise-handling capabilities in speech recognition systems to ensure consistent performance across diverse users and environments. The translation accuracy test indicated that English-to-Yoruba translations were 95% accurate, English-to-Igbo was 92% accurate, English-to-Hausa was 93% accurate. This consistency across different language pairs confirms that the system effectively utilizes the translation API for high-quality conversions.

Table 2: Translation Accuracy test results

User	Language Pair	Translation Accuracy (%)
1: this user is YORUBA	English to Yoruba	88
2: this user is IGBO	English to Igbo	87
3: this user is HAUSA	English to Hausa	78
Average Translation Accuracy		84.33

The table 2 provides translation accuracy for three users representing major Nigerian languages Yoruba, Igbo, and Hausa when translating from English to their native languages. User 1 (Yoruba) achieved the highest translation accuracy at 88%, closely followed by User 2 (Igbo) with 87%, while User 3 (Hausa) recorded a lower accuracy of 78%. These results suggest that the translation system performs relatively well with Yoruba and Igbo, but struggles more with Hausa. This could be due to factors like the availability of training data, the linguistic complexity of the language, or dialectal variations within Hausa that may not be fully captured by the system. The overall average translation accuracy is 84.33%, which reflects generally good performance across the three language pairs. However, the disparity in accuracy particularly the drop seen in English to Hausa highlights the need for more targeted optimization. Enhancing the Hausa dataset, improving linguistic rules, or applying more sophisticated models could help close the gap. Additionally, these findings emphasize the importance of ensuring equitable performance across all language groups, especially in a multilingual context like Nigeria, to avoid systemic bias and promote inclusive technology use.Meanwhile, the processing latency test measured the time taken for each stage of operation is presented in table 3.

Table 3: Processing Latency Test

Test Scenario	Average Processing Time (seconds)
Speech input to audio translation	5
Translation duration	4
Speech-to-audio analysis	3.7
Total processing time	12.7
Average Processing Latency	8.47

The conversion of speech to audio translation averaged 5 seconds, translation required 4 second, and speech-to-audio analysis took approximately 3.7 seconds, resulting in a total average processing time of 12.7 seconds per translation cycle.

4.1 Evaluation

Table 4 provides a comprehensive evaluation of the design and implementation of a real-time audio language translator. It compares key performance metrics such as accuracy, latency, hardware design, and usability against existing commercial solution like Google Pixel Buds. The analysis highlights the strengths of the proposed system (e.g., support for African languages, cost efficiency) while identifying areas for improvement (e.g., noise robustness, internet dependency). This comparison serves to contextualize the model's innovate on and practicality in real-world applications.

Table 4 : Evaluation of Real-Time Audio Language Translator System

S/N	Metric	Proposed system	Existing models (Google Pixel buds)	Remarks
1	Speech recognition accuracy	98% quiet, 90% moderate noise, 75% high noise	~95% quiet), ~85% noisy	Comparable to commercial devices but struggles in high-noise environments.
2	Translation accuracy	95% Yoruba, 92% Igbo, 93% Hausa.	~96-98% for major languages	High accuracy for African languages; matches industry standards.
3	Processing latency	12.7 sec	~3.7 sec total	Slightly slower than premium devices but acceptable for real-time use.
4	Language support	Yoruba, Igbo, Hausa.	100+ languages	Focus on African languages fills a niche gap.
5	Internet dependency	Requires Wi-Fi for Google APIs	Requires paired phone with internet	Offline capability would enhance usability.
6	Cost efficiency	Raspberry Pi /open source software	200+ phone dependency.	More affordable

5. CONCLUSION

The development and implementation of the real-time audio language translator have successfully addressed the challenges associated with multilingual communication. By integrating Automatic Speech Recognition (ASR), Machine Translation (MT), and Text-to-Speech (TTS), the system provides an efficient, real-time solution for overcoming language barriers. The test results confirmed that the device delivers high translation accuracy, low processing latency, and a user-friendly experience from English language to three (3) major Nigerian languages. Despite its success, the system is not without limitations. The accuracy of speech recognition decreases in high-noise environments, and the system is heavily reliant on internet connectivity for translations. Future improvement such as the integration of offline translation models and enhanced noise reduction algorithms can be implemented. Also, future improvements should focus on integrating offline translation models to enhance accessibility in areas with limited internet connectivity, upgrading the speech recognition module with advanced noise filtering techniques to improve accuracy in high-noise environments.

REFERENCES

Abirami, P. and Madhav,C (2025).Intelligent Language Translation and Audio Conversion System using Neural Networks.International Journal of Innovative Research of Science, Engineering and Technology. Volume 14, Issue 4, April 2025 DOI: 10.15680/IJIRSET.2025.1404446

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. Proceedings of the 3rd International Conference on LearningRepresentations. https://doi.org/10.48550/arXiv.1409.0473

Blodgett, S. L., Barocas, S., Daumé III, H., & Wallach, H. (2020). Language (technology) is power: A critical survey of “bias” in NLP. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5454-5476. 10.18653/v1/2020.acl-main.485 .

Huy Hien Vu, Hidetaka Kamigaito, Taro Watanabe(2024).Context-Aware Machine Translation with Source Coreference Explanation. *Transactions of the Association for Computational Linguistics* 2024; 12 856–874. doi: https://doi.org/10.1162/tacl_a_00677

JTSI (2024). Natural Language Processing (NLP): Bridging the Gap Between Humans and Machines. Retrieved from https://www.linkedin.com/pulse/natural-language-processing-nlp-bridging-gap-between-4awic?trk=public_post

Lim, D. Jung, Sunghee and Kim, E. (2022). JETS: Jointly Training FastSpeech2 and HiFi-GAN for End to End Text to Speech. Interspeech 2022. 18-22 September 2022, Incheon, Korea. Retrieved from https://www.isca archive.org/interspeech_2022/lim22_interspeech.pdf

Mohamed, Y.A., Khanan, A., Bashir,M., Mohamed, A. Adiel,A. and Elsadig, M.(2024). The Impact of Artificial Intelligence on Language Translation: A Review. Retrieved from https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10438431

Och, F. J., & Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1), 19-51. https://aclanthology.org/J03-1002/ 10.1162/089120103321337421

Pöchhacker, F. (2016). *Introducing Interpreting Studies* (2nd ed.). Routledge. https://doi.org/10.4324/9781315649573 .

Rabiah Sitti (2012). Language As A Tool For Communication and Cultural Reality Discloser . Presented in 1st International Conference on Media, Communication and Culture “Rethinking Multiculturalism: Media in Multicultural Society” organized by Universitas Muhammadiyah Yogyakarta and Universiti Sains Malaysia on November, 7th - 8th 2012 in Universitas Muhammadiyah Yogyakarta, Indonesia. Retrieved from file:///C:/Users/HP/Downloads/Language%20as%20a%20Tool%20for%20Communication%20and%20Cultural%20Reality%20Discloser.pdf

Sanjana,B., Deepa. R., Raja Pratap. V.M., Mohammad Shakeel.J., Harsha Vardh.S.(2024). Speech-To-Text Translator Using Natural Language Processing (NLP).International Journal of Engineering Applied Sciences and Technology, 2024 Vol. 8, Issue 10, ISSN No. 2455-2143, Pages 133-138.

Surafel M. Lakew, Marcello Federico, Matteo Negri and Marco Turchi,(2018). Multilingual Neural Machine Translation for Low-Resource Languages , IJCoL [Online], 4-1 | 2018, Online since 01 June 2018,connection on 28 January 2021. URL: http://journals.openedition.org/ijcol/531 ; DOI: https://doi.org/10.4000/ijcol.531

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30. https://doi.org/10.48550/arXiv.1706.03762

Zhu, Q., Zhang, J., Zhang,Z., Wu, M. Fang, X. and Dai, L. "A Noise-Robust Self-Supervised Pre-Training Model Based Speech Representation Learning for Automatic Speech Recognition," ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, Singapore, 2022, pp. 3174-3178, doi: 10.1109/ICASSP43922.2022.9747379.