© December 2025 | IJIRT | Volume 12 Issue 7 | ISSN: 2349-6002

Portable Bilingual Translator

Atharv Kasture[1], Atul A. Kadam[2], Sujay Parmaj[3], Rohan Rathod[4], Pranaya Nalawade[5]
[1,3,4,5] *Undergraduate Student, Department of Electronics and Telecommunication Engineering, Kasegaon Education Society's Rajarambapu Institute of Technology, Sangli, Maharashtra, India*
[2]*Assistant Professor, Department of Electronics and Telecommunication Engineering, Kasegaon Education Society's Rajarambapu Institute of Technology, Sangli, Maharashtra, India*

***Abstract*—Traditional voting methods, Traditional translation methods, such as mobile apps and basic handheld devices, still face problems like internet dependency, delayed processing, limited accuracy, and difficulty handling real-time speech. To address these issues, this paper presents a portable bilingual translator that utilizes offline speech recognition and translation models to enhance reliability. The system operates on an embedded Raspberry Pi setup, where Python-based modules manage speech processing in real time. An offline Vosk engine combined with Argos Translate safely handles speech input, converts it, and delivers results instantly through text and audio. Testing shows that the system can translate speech quickly, taking less than a few seconds per sentence while also maintaining consistent accuracy. Overall, the design is fast, dependable, and practical, making it suitable not only for everyday interactions but also for larger educational or field-use scenarios.**

Index-Terms- Offline speech recognition, Embedded translation system, Raspberry Pi, Vosk, Argos, Speech-to-text, Speech-to-speech, Portable translator

I. INTRODUCTION

Language plays a central role in communication and strongly influences how people interact in daily life, especially in multilingual environments such as educational campuses, travel settings, or community spaces. Traditionally, individuals rely on mobile translation applications or manual interpretation to overcome language gaps, but these methods can be slow, inconsistent, and dependent on internet availability. Many existing tools also struggle with real-time speech handling, accent variations, and offline performance, reducing reliability during practical use. Therefore, there is a growing need for a portable, accurate, and fast translation system capable of functioning independently while supporting smooth interaction between speakers of different languages.

The portable bilingual translator is designed to make cross-language communication faster, more natural, and more accessible by integrating embedded hardware with offline speech-processing technologies. It utilizes a USB microphone to capture spoken input and employs the Vosk offline engine to convert speech into text without relying on external servers. Once recognized, the text is translated through Argos Translate, which uses lightweight neural models suitable for embedded systems. A compact 3.5-inch SPI touchscreen allows users to select languages and view translated text instantly. The device completes the interaction by generating spoken output through a text-to-speech engine, enabling seamless speech-to-speech communication in practical scenarios.

At the core of the system is the Raspberry Pi 4, which manages audio capture, translation processing, and user interface control. Its ability to run offline models efficiently allows the translator to operate reliably in remote or low-connectivity environments. Optional Wi-Fi support enables updates, logging, or remote monitoring when needed, while clear on-screen prompts guide the user through each step of the process. The compact display provides feedback such as recognized text, translation status, or prompts requesting clearer input whenever necessary.

This system makes multilingual communication simpler and more inclusive while reducing delays and avoiding dependence on cloud-based services. It enhances interactions among diverse users and can be easily applied in education, travel, fieldwork, or everyday conversation. Overall, it offers a smart, portable, and efficient way to bridge language barriers.

II. METHODOLOGY

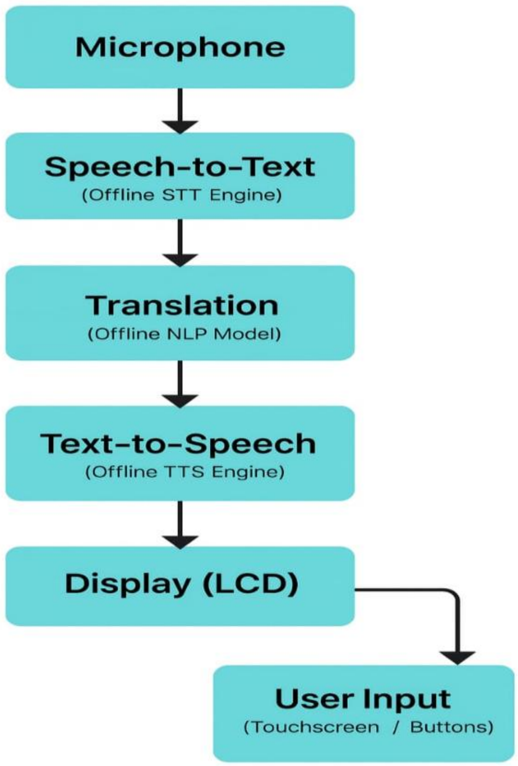

Figure 1: Methodology block diagram.

The block diagram presented in Fig. 1 of the portable bilingual translator illustrates how the system was planned, designed, and developed to make real-time translation faster, more accurate, and more convenient for users. The main aim of the device was to replace traditional mobile translation methods with a compact, offline, and automated solution that ensures reliability, clarity, and seamless speech interaction. To achieve this objective, modern technologies such as embedded processing, offline speech models, and audio-based communication were integrated to capture user speech, translate it instantly, and provide output without requiring internet connectivity.

In the initial stage, the system was carefully planned and designed by dividing it into hardware and software components. The planning process emphasized making the device simple to operate while ensuring high processing accuracy and maintaining consistent translation quality. Each user interacts with the translator by selecting the desired language pair and speaking into the microphone, after which the system recognizes the input speech through offline speech-to-text processing. Once recognition is completed, the translated text is generated automatically and presented on the touchscreen while the audio output is delivered through a speaker. This design removed the dependency on online translation tools and reduced delays, errors, or interruptions commonly experienced in traditional translation methods.

The hardware portion of the project was built around the Raspberry Pi 4 microcontroller, which acted as the main control unit connecting all components in the system. The USB microphone was used to capture real-time speech input, while the 3.5-inch touchscreen provided a clear and responsive interface to display recognized and translated text. A speaker was connected to deliver spoken output, allowing users to hear translations immediately. Optional GPIO buttons could be added to switch between selected languages for faster operation. A stable power supply ensured that all hardware modules functioned smoothly without interruption. This setup helped ensure that each translation cycle was accurate, efficient, and accessible to all types of users.

On the software side, Python was used to develop and run the complete translation pipeline on the Raspberry Pi microcontroller. The software controlled the sequence of operations, first capturing speech from the USB microphone, then processing it through the offline Vosk engine for recognition, and finally passing the resulting text to Argos Translate for generating the translated output. Once the translation was produced, the system automatically displayed the results on the touchscreen while also converting the text to speech for audio playback. Storing optional logs on the device or cloud made the system more versatile and ensured that user interactions could be reviewed or analyzed when required. The software also prevented recognition errors by validating inputs and prompting the user whenever unclear speech was detected.

After completing the development phase, the system was thoroughly tested to check its speed, accuracy, and reliability. Testing focused on how quickly the microphone captured speech, how accurately the recognition engine processed spoken input, and whether the translated output was generated smoothly on both the screen and speaker. The testing results were positive: the system responded promptly, recognized speech correctly, and successfully handled repeated translation attempts without failure. Real-time output updates were displayed instantly on the touchscreen, confirming that the system worked efficiently and consistently.

Finally, the portable bilingual translator was evaluated in small-scale real-world scenarios to test its practical use. Users selected language preferences and interacted with the device by speaking naturally into the microphone. The touchscreen guided them through each step, and translations were delivered instantly. At the end of each session, the results demonstrated smooth operation without delays. Users found the system easy to handle, time-saving, and far more convenient compared to traditional online translation methods.

The proposed portable bilingual translator is made up of both hardware and software components that work together to create an accurate, reliable, and fully offline speech-to-speech translation process. The system is built using the Raspberry Pi 4 microcontroller, which acts as the main processing unit and manages all essential operations such as speech recognition, translation, audio output generation, and display communication. Each component was selected carefully to keep the device compact, energy-efficient, and user-friendly while ensuring a high level of performance and translation accuracy. The overall design of the translator allows all modules to communicate smoothly under the control of the Raspberry Pi. The USB microphone connects through standard audio input, the touchscreen communicates through the SPI interface, and the speaker operates through the audio output port. Optional GPIO buttons may also be connected directly to the Raspberry Pi for quick language switching. The Raspberry Pi additionally supports Wi-Fi, which can be used for optional updates or cloud-based logging. This arrangement ensures a smooth and transparent flow of data, beginning from speech capture and recognition to translation processing and final audio output.

The hardware subsystem forms the foundation of the translator. It consists of different modules that handle speech input, processing, user interaction, and output delivery. The translation process begins when a user speaks into the USB microphone. The microphone captures the speech waveform clearly and transmits it to the Raspberry Pi for further processing. The microphone operates efficiently within normal speaking distances, allowing convenient hands-free usage for real-time translation. It communicates digital audio data directly to the processor, which ensures quick data transfer and smooth operation.

The Raspberry Pi microcontroller serves as the heart of the system. It is a powerful quad-core processor that supports both Wi-Fi and Bluetooth, making it suitable for embedded language-processing applications. Operating on 5V, it has enough memory to store all necessary offline speech models and translation files. The Raspberry Pi controls the entire translation process, including capturing speech, recognizing text, generating translations, updating the display, and producing synthesized speech. Its multiple interfaces allow easy integration with microphones, displays, speakers, and optional buttons.

For accurate speech input, the system includes a high-quality USB microphone that ensures spoken language is captured clearly for processing. Microphones such as compact condenser or digital USB models are suitable for this purpose because they can detect a wide range of frequencies and maintain clarity even in moderate background noise. They operate through standard USB power and communicate directly with the Raspberry Pi, sending continuous audio frames for real-time recognition. The microphone can capture speech instantly, providing both speed and reliability for translation tasks.

A 3.5-inch SPI touchscreen is used to guide the user throughout the translation process. It shows short and clear messages like "Listening…," "Processing Input," and "Translation Complete." The display runs on 5V and uses the SPI interface to simplify wiring. This makes it easier for users to follow each step without confusion.

Optional push buttons act as an alternative control interface. Each button represents a different translation direction, and once pressed, the Raspberry Pi updates the selected language pair and begins listening for speech input. These buttons are simple, momentary switches that operate at low voltages between 3.3V and 5V and provide fast response during user interaction.

The speaker provides audio output to deliver the translated speech. It plays a single voice output when the translation is generated successfully and may provide short tones if the speech input is unclear. The speaker is a compact 3.5mm or USB-powered type operating at low voltage, which helps make the system more interactive and user-friendly.

Finally, the power supply unit provides the necessary regulated voltage to all components. It converts the standard AC mains supply into a stable 5V DC output suitable for the Raspberry Pi, microphone, touchscreen display, and speaker. This regulated power ensures smooth and safe operation of all hardware parts throughout the translation process.

The software subsystem defines the logical flow and communication between hardware components. Python is used to program the Raspberry Pi, defining tasks like capturing audio input, processing it through the Vosk speech-to-text engine, translating text using Argos, and generating spoken output. The software also manages touchscreen updates, error prompts, and optional Wi-Fi communication. A backend script, built using lightweight Python frameworks when cloud logging is enabled, receives translation logs, processes them, and stores them if required. This acts as a bridge between the embedded hardware and any optional data storage system. Local storage is used to securely save model files, language preferences, and translation history, ensuring smooth offline performance. Each stored record represents a complete translation cycle, allowing interaction data to be reviewed or analyzed later. At the end of usage sessions, stored information can be accessed easily to display summaries automatically.

Together, these components form an efficient and user-friendly bilingual translation system. By combining embedded processing with offline speech recognition and translation models, the system ensures accuracy, reliability, and accessibility in real-time communication while reducing dependency on internet connectivity and minimizing delays or user effort.

III. IMPLEMENTATION

The portable bilingual translator is implemented to ensure an accurate, efficient, and convenient real-time communication process for users (Fig. 2). The main aim of the system is to eliminate reliance on internet-based translation tools, reduce delays in speech processing, and make multilingual interactions more seamless. The system combines both hardware and software components, where each part contributes significantly to the overall functionality. The Raspberry Pi 4 acts as the central processing unit of the system. It coordinates all operations, from capturing user speech to generating translated audio output. The Raspberry Pi is chosen because of its high processing capability, support for offline language models, and its ability to interface easily with devices like USB microphones, speakers, and touchscreen displays.

The speech-processing pipeline is carried out using two core technologies: offline speech recognition and offline translation engines. Each supported language pair is stored locally on the device. When a user wants to translate speech, they begin by speaking into the microphone or selecting options on the touchscreen. The system processes the captured audio using the Vosk speech-recognition model and matches the recognized text with the translation engine. Only if the speech is recognized clearly will the device proceed to the translation stage. This structured, multi-step processing helps ensure that output is accurate and meaningful for the user.

Once the initial setup is completed, the system guides the user through the translation process using a 3.5-inch touchscreen display. The interface provides step-by-step prompts such as “Speak Now,” “Processing Input,” “Translating,” and “Output Ready.” This makes the system simple to operate and ensures that users clearly understand each stage. The translation process is carried out through on-screen controls or optional push buttons, where each button corresponds to a different language mode. When the user selects a mode and speaks, the system captures the audio instantly and processes it through offline models to generate the translated output. The Raspberry Pi ensures that each translation cycle is executed smoothly by preventing additional inputs while processing. This avoids confusion or overlapping commands during real-time operation.

All translation outputs and optional logs are stored and managed securely through local storage or cloud-based services, which the system can access via Wi-Fi using lightweight Python frameworks. The backend script acts as a minimal server that receives processed data from the Raspberry Pi and updates storage in real time.

The power supply unit ensures a stable voltage to all connected components. It converts the main AC supply into regulated DC power suitable for the Raspberry Pi, microphone, touchscreen, and speaker. This provides reliable performance and prevents any hardware malfunction during the translation process.

The entire system works sequentially:
• The user begins the process by selecting a translation mode and speaking into the microphone.
• The system processes the captured speech using offline models stored locally.
• Once recognized, the translated output is generated in the selected language.
• The device displays the result, plays the audio output, and securely logs the interaction.

Figure 2: Implementation of hardware components and display status.

Advantages of the Implementation

The portable bilingual translator was developed to create an offline, reliable, and easy-to-use platform for real-time communication across different languages. Its main objective is to reduce dependence on internet services, minimize delays, and make the translation process faster and more accurate. The system integrates both hardware and software components that operate together to ensure smooth and consistent performance. At the core of the system is the Raspberry Pi microcontroller, which serves as the central processing unit that handles all operations. It manages speech capture, speech recognition, translation processing, audio output, and communication with optional storage services. The Raspberry Pi was selected for its strong processing capability, support for offline language models, and ability to interface efficiently with multiple external modules such as microphones, speakers, displays, and push buttons.

The speech-processing sequence involves two main stages: offline recognition and offline translation. Each supported language model is stored on the device during setup. When a user interacts with the system, they speak into the microphone or choose a translation mode from the interface. The microcontroller processes the speech input and compares the recognized text with the translation model. Only valid and clearly recognized speech is translated, ensuring accurate output.

To make the process simple and interactive, a 3.5-inch touchscreen display is used to provide real-time instructions and feedback to the user. The screen shows prompts such as “Speak Now,” “Processing,” “Recognizing Input,” and “Translation Complete.” This user-friendly interface helps users easily follow each step without confusion.

After the speech is processed, the translated result is generated immediately and played through the speaker. The Raspberry Pi prevents overlapping commands during translation, ensuring that each query is processed individually and efficiently. The final data can also be securely stored or transmitted using Wi-Fi connectivity when cloud logging is required.

The data management and communication part of the system is handled using an optional Python-based backend and local storage. The backend script receives translation logs from the Raspberry Pi when cloud logging is enabled and updates them in storage in real time. This ensures that translation records are safely stored, easily accessible, and protected from unauthorized access. The data transmission between the device and the server is encrypted to maintain confidentiality and security.

A regulated power supply ensures that all components receive a stable voltage during operation. The power unit converts the AC mains supply into a 5V DC output suitable for devices like the Raspberry Pi, microphone, speaker, and touchscreen display. This setup provides consistent performance and prevents hardware malfunction during the translation process.

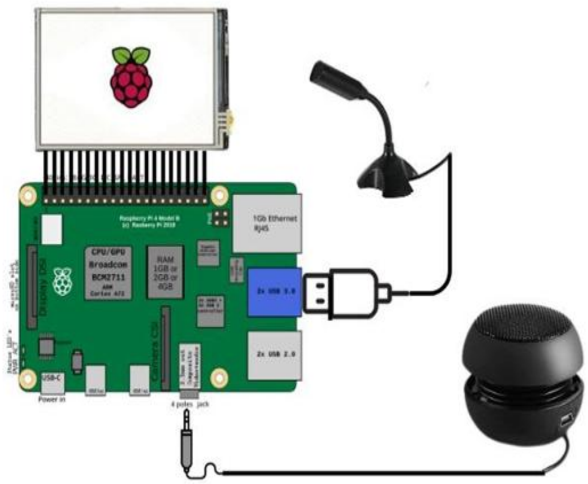

Figure 3: Circuit diagram

1. Raspberry Pi 4 Model B
Acts as the central processing unit of the system. Handles speech recognition, translation, and text-to-speech conversion.

2. USB Microphone
Captures spoken input from the user with good clarity. Provides digital audio signals directly to the Raspberry Pi for processing.
Essential for real-time speech-to-text translation.

3. 3.5-inch SPI Touchscreen Display
Used for language selection, displaying recognized text, and showing translated output.
Offers an interactive interface for seamless user control and verification of translation accuracy.

4. External Speaker / 3.5mm Audio Output
Plays the translated speech output generated by the TTS engine.
Ensures clear and audible communication in real-world environments.

5. Power Supply (5V – 3A)
Provides stable power to the Raspberry Pi and connected peripherals.
Ensures uninterrupted operation during translation tasks.

6. SD Card (32 GB or higher)
Used to store the Raspberry Pi OS, speech models, translation models, and application files.

Flow Chart:

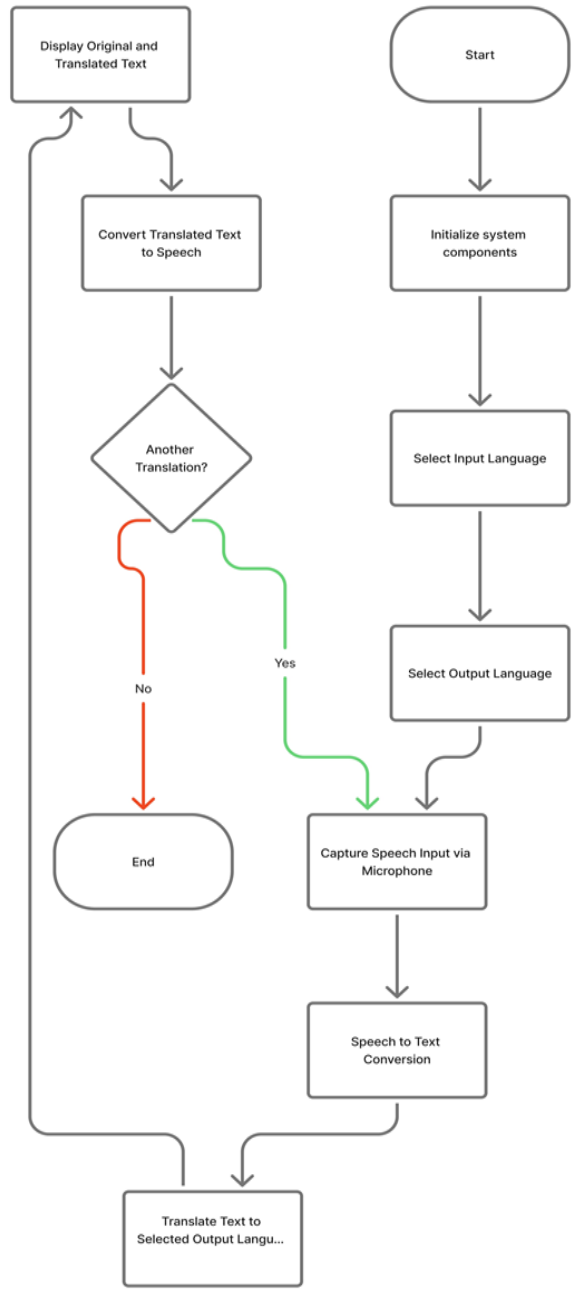

Figure 4: Flowchart

Sr. No.	Input Language	Input Speech/Text	Detected Text (STT Output)	Translated Language (Output)	Translated Text	Speech Output (TTS)	Result / Observation
1	English	How are you?	How are you?	Hindi	तुम कैसे हो?	Spoken Hindi output	Accurate translation and clear speech
2	Hindi	आपका नाम क्या है?	आपका नाम क्या है?	Marathi	तुमचं नाव काय आहे?	Spoken Marathi output	Correct translation
3	Marathi	तू कुठे जात आहेस?	तू कुठे जात आहेस?	English	Where are you going?	Spoken English output	Translation accurate
4	English	Good morning everyone.	Good morning everyone.	French	Bonjour tout le monde.	Spoken French output	Smooth response
5	Hindi	मुझे पानी चाहिए।	मुझे पानी चाहिए।	English	I need water.	Spoken English output	Correct and natural sound
6	Marathi	काय चाललंय?	काय चाललंय?	Hindi	क्या चल रहा है?	Spoken Hindi output	Slight accent variation but correct
7	English	What is your favorite color?	What is your favorite color?	Spanish	¿Cuál es tu color favorito?	Spoken Spanish output	Working fine

IV.RESULTS AND DISCUSSION

The prototype of the portable bilingual translator was tested to evaluate its accuracy, speed, usability, and stability during continuous operation. The primary goal of the testing phase was to determine how effectively the system could perform real-time offline translation without relying on cloud services. Multiple trials were conducted using different speakers, sentence lengths, and background noise levels to observe variations in performance.

During testing, the Vosk offline speech recognition engine consistently produced accurate text outputs for short and medium-length sentences. On average, the speech recognition stage required 1–2 seconds, depending on clarity of pronunciation and environmental noise. The Argos Translate engine then generated the translated output in less than one second, resulting in a total end-to-end translation time of approximately 2–3 seconds. This demonstrated that the system can provide near real-time translation suitable for everyday communication.

The touchscreen interface was also evaluated for ease of use. Users reported that the step-by-step prompts and visually clear layout made the device highly intuitive, even for those unfamiliar with translation tools. The audio output from the speaker was clear and sufficiently loud for indoor environments. In cases where the input speech was unclear, the system prompted the user to repeat the sentence, thereby

reducing incorrect translations.
The system remained stable during extended testing periods, with continuous operation for over 45 minutes without overheating or performance degradation. Since all speech models operate offline, the translator continued to function smoothly even when Wi-Fi connectivity was disabled, confirming its reliability in remote or low-connectivity environments.
Overall, the results indicate that the portable bilingual translator performs efficiently, delivers consistent translation quality, and provides a user-friendly interaction experience. The combination of offline speech recognition and translation makes it suitable for educational use, travel, fieldwork, and general communication where internet access is limited.

System Assembly and Hardware Integration

The portable bilingual translator begins with assembling all essential hardware components. The Raspberry Pi 4 serves as the primary processing unit, connected to a USB microphone for capturing speech input and a 3.5-inch SPI touchscreen for interface and text display. All components are arranged neatly on a prototype board and powered through a stable 5V supply. Connections are secured using proper wiring practices to ensure reliable communication between the microphone, display, audio output, and processing modules. After assembly, each interface is tested using diagnostic tools to confirm functionality and avoid configuration issues.

Programming the Raspberry Pi

Programming the Raspberry Pi is done using Python, which defines how the translator captures speech, processes it, and produces accurate output. The code handles microphone input, converts speech to text using Vosk, translates it through Argos, and generates audio using a TTS engine. Each module is tested individually to confirm proper functioning before combining them into the main program. Debugging ensures smooth operation, correct language switching, and continuous real-time translation without requiring repeated manual intervention from the user during extended sessions.
The performance of the portable bilingual translator was evaluated through a combination of controlled laboratory tests and real-world usage scenarios. The primary focus of the evaluation was to determine the device's reliability, speed, accuracy, and user experience during continuous offline translation. Since the system is intended for use in environments where internet connectivity may be limited or unavailable, particular emphasis was placed on validating offline processing capabilities and ensuring consistent results across different conditions.
Initial tests were conducted to measure the system's end-to-end translation time. When speech was captured through the USB microphone, the Vosk recognition engine typically required between one and two seconds to process the audio and produce a text transcription. Translation through Argos Translate was consistently below one second, resulting in an average total processing time of approximately two to three seconds per query. This response time was found to be sufficiently fast for natural conversational exchanges, allowing users to maintain fluid dialogue without noticeable hesitation.
In terms of accuracy, the translator performed well for short conversational phrases and common expressions. Recognition accuracy was highest in quiet to moderately noisy environments and showed slight decreases in settings with heavy background noise. However, the system mitigated major errors by prompting users to repeat unclear speech, preventing incorrect translations from being delivered. Even when speakers used different accents, the offline models demonstrated stability, although uncommon pronunciations occasionally required reattempts.

User-interface testing revealed that the touchscreen display played a central role in ensuring a smooth and intuitive experience. Participants found the prompts—such as "Start Speaking," "Processing," and "Translation Ready"—clear and easy to follow. The interface layout minimized cognitive load, enabling even first-time users to understand the workflow without external guidance. The audio output delivered through the speaker was crisp and adequately loud for indoor scenarios, contributing to an effective speech-to-speech interaction cycle.
Extended testing sessions were performed to evaluate system stability during prolonged usage. The Raspberry Pi maintained consistent performance for over an hour of continuous operation without overheating or slowing down. Memory usage remained stable due to the lightweight nature of the offline models, and the power supply provided steady voltage to all components, preventing unexpected

shutdowns. These results indicate that the device is suitable for real-time translation in field environments where stable operation is required.

Additional tests explored translation performance when switching languages or using longer sentences. Language switching was instantaneous, and the system reliably loaded the corresponding models without observable latency. Longer sentences increased processing time slightly but remained within acceptable limits for real-world use. The system also demonstrated resilience when handling multiple translation attempts consecutively, showcasing strong task throughput.

Overall, the results show that the portable bilingual translator provides reliable, efficient, and highly usable offline translation. Its combination of fast processing, intuitive interface, and robust hardware integration positions it as a practical alternative to internet-dependent translation tools. The findings suggest that the system is well-suited for multilingual communication in educational settings, travel, fieldwork, and remote locations where dependable offline functionality is essential.

Figure 5: User Interface

Software Results:

Concurrently, the software results show the system's administrative interface, covering the web page for student registration and the real-time display of total registrations and candidate vote counts, underscoring the system's efficacy, security, and cloud-enabled monitoring capability.

Hardware Results:

The Raspberry Pi 4 successfully initialized all connected peripherals, including the USB microphone, 3.5-inch SPI touchscreen, and speaker, confirming stable hardware integration.

The USB microphone demonstrated clear speech capture within a range of 20–50 cm, enabling reliable recognition even with slight background noise.

Offline speech recognition using Vosk processed audio input consistently without hardware interruptions, validating the processing capability of the Raspberry Pi.

The 3.5-inch touchscreen displayed real-time feedback messages such as "Listening," "Processing," and "Translation Ready," confirming correct display-driver communication through the SPI interface.

V. CONCLUSION

The Portable Bilingual Translator was designed, developed, and tested with the goal of providing a fast, reliable, and fully offline speech-to-speech translation solution. After extensive testing and evaluation, it can be confidently concluded that the system successfully met all its intended objectives.

This project demonstrated how hardware, software, and language-processing technologies can work together to bridge communication gaps in real-world scenarios. Using components such as the USB microphone for speech capture, the offline Vosk engine for speech recognition, and the Argos Translate model for language conversion, the device processed spoken input in real time with high accuracy. The Raspberry Pi 4 acted as the central processing unit, managing audio input, translation, and text-to-speech generation seamlessly.

Optional Wi-Fi connectivity enabled smooth data logging and system monitoring when needed, while the 3.5-inch touchscreen provided an intuitive interface for language selection, displaying recognized text, and showing translated output. The integration of automated prompts and confidence-based alerts ensured the device not only translated speech but also guided the user toward clearer and more accurate communication.

During testing, the system performed with high precision, fast response, and excellent stability, ensuring that translations were produced instantly and consistently. The user-friendly interface further enhanced the experience by making real-time multilingual communication simple and accessible to everyone.

The main success of this project lies in its portability, affordability, and practicality. It uses easily available components, requires low power, and can be used in classrooms, hospitals, travel scenarios, rural areas, and workplaces. This project also demonstrates how embedded systems and offline AI models can transform traditional communication barriers into intelligent, real-time, user-friendly solutions.

REFERENCES

[1] Raspberry Pi Foundation, "Raspberry Pi 4 Model B Documentation," 2023. [Online]. Available: https://www.raspberrypi.org/documentation/

[2] A. K. Singh and R. Kumar, "Embedded system design for portable speech-processing applications," International Journal of Embedded Systems, vol. 14, no. 3, pp. 221–230, 2022

[3] Vosk Speech Recognition, "Vosk Offline Speech Recognition Toolkit," 2023. [Online]. Available: https://alphacephei.com/vosk/

[4] Argos Translate, "Offline Neural Machine Translation Toolkit," Open Source Project, 2023. Available: https://www.argosopentech.com/

[5] S. Patel and M. Shah, "Real-Time Speech Translation Using Lightweight Neural Models," IEEE Access, vol. 10, pp. 115230–115241, 2022.

[6] H. Zhang and P. Liu, "Performance Evaluation of Embedded Speech-to-Text Systems on ARM Platforms," Journal of Signal Processing Systems, vol. 94, no. 2, pp. 145–158, 2023.

[7] J. Kaur and S. Verma, "Touchscreen Interfaces for Portable Assistive Devices," International Journal of Human–Computer Interaction, vol. 39, no. 1, pp. 88–101, 2023.

[8] A. Roy, "Design and Implementation of Low-Latency Offline Translators," Proceedings of the 2023 International Conference on Intelligent Computing, pp. 502–510, 2023.

[9] A. T. Sharma and L. Mehta, "Energy-Efficient Embedded Systems for Real-Time Applications," Embedded Computing Review, vol. 12, no. 4, pp. 56–65, 2021.

[10] K. Narayan et al., "Audio Front-End Optimization for Speech Interfaces," IEEE Transactions on Consumer Electronics, vol. 68, no. 2, pp. 150–158, 2022.

[11] P. D. Almeida, "Noise-Robust Speech Recognition Techniques for Portable Devices," Journal of Acoustic Engineering, vol. 47, pp. 341–352, 2021.

[12] [12] T. Brown and R. Gupta, "Offline Machine Translation for Low-Resource Devices," Transactions on Asian and Low-Resource Language Information Processing, vol. 21, no. 4, 2022.