



REAL TIME TEXT TRANSLATOR

¹Abin John Jinoy, ²Adwaith K Raman, ³Christin K Y, ⁴Hafez Abid, ⁵Dr. Sandeep Bhat, ⁶Sudheesh K P
1,2,3,4 Students, 5 Professor, 6 Assistant Professor.
1,2,3,4,5,6, Computer Science and Engineering.

1.2.3.4.5.6, Srinivas Institute of Technology, Mangalore, India

Abstract: Real time text translator App uses Google Translate API to translate entered text or message into a selected language. For instance, if the user entered text in English & translation language is selected as Marathi, the google translation API will translate it into the Marathi Language. The recognition engine for the Neural Machine Translation (NMT) model in the Google Translation API supports more than a hundred languages. Language code parameters are used to specify these languages within a recognition request. Users can switch the language; the user can enter the text in Hindi or Marathi or any other language. There is another option, if the user speaks any text or message, it will detect it and translate it into the selected language.

Index Terms – Deep Learning, CNN, OpenMP, VGG16.

I. INTRODUCTION

In today's increasingly interconnected world, globalization has significantly amplified the need for seamless communication across diverse linguistic and cultural boundaries. Despite advancements in technology, language barriers remain a significant obstacle that hampers personal, educational, and professional interactions on a global scale. To address this pressing challenge, the proposed project aims to design and develop a comprehensive text translation application that facilitates accurate, fast, and user-friendly translations between multiple languages. The core objective is to bridge the communication gap by enabling users to input text in English alphabets and receive real-time translations in their chosen target language. This will be achieved through the integration of cutting-edge machine learning algorithms and natural language processing (NLP) techniques, which ensure that the translations are not only grammatically correct but also contextually appropriate.

The application will feature a clean and intuitive user interface, allowing users to input text easily, select their desired output language, and view translated content instantly. It will support multi-language translation, enabling interactions among users from various linguistic backgrounds. The translation functionality will be powered either by robust third-party APIs such as Google Translate or through a custom-built neural machine translation (NMT) engine designed to improve translation quality over time via continuous learning. Additionally, the system will be designed to deliver high performance with minimal latency, providing users with faster and more responsive translations compared to many existing tools.

From a technical perspective, the application will be developed using Python for backend processing, JavaScript and HTML/CSS for the frontend interface, and will be deployed on Android platforms using Android Studio, with XML and Java JDK 8.0 for mobile development. The backend infrastructure will utilize MySQL 5.0 as the database and PHP 5.6.31 for server-side scripting, ensuring data management is secure and scalable. Genymotion 2.12.0 will be used as the primary emulator for development and testing, and the app will be optimized to run on Android version 4.2 and above, ensuring broad device compatibility.

The application is intended to be lightweight yet powerful, requiring only modest hardware specifications—namely, a 1.5GHz Quad-Core processor, at least 1 GB of RAM, and 16 GB of internal storage. The solution will be offered free of charge, making it accessible to a wide range of users, including students, travelers, business professionals, and language learners. By enhancing usability, scalability, and translation accuracy, the project aims to make a tangible impact in breaking down language barriers, thereby promoting inclusivity and enhancing cross-cultural communication on a global scale.

II. PROPOSED METHODOLOGY

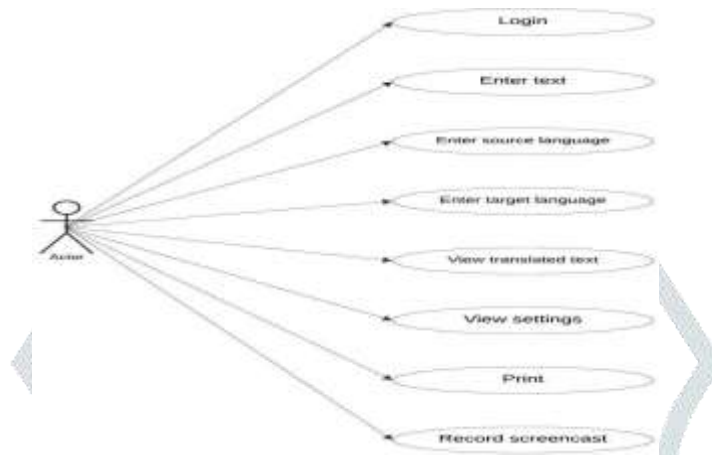


Fig.2.1: The proposed system.

The suggested system's architecture is depicted in the diagram 1. A web application will be used by the user to communicate with the system. It shows the interaction between user and the system. User interacts with the system by selecting languages, entering text, clicking translate, and viewing the result. The system translates the text and displays the result or an error if something goes wrong. This use case diagram effectively captures the functionality of your translation tool, focusing on the key interactions between the user and the system. The figure 2 Shows the flow diagram of model operations.

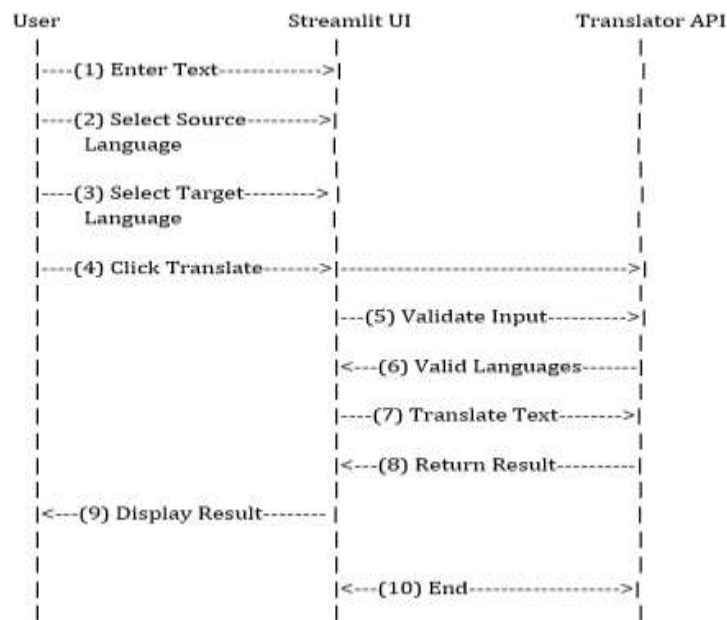


Fig.2.2: Model operations flow diagram.

In a real-time text translator application, the user inputs text through the interface, which is processed and sent to a language detection module. The detected language and text are then passed to the translation engine (e.g., Google Translate or a custom model) for translation. The translated text is formatted and displayed back to the user. Data stores may be used to save preferences or frequently used phrases to enhance performance and accuracy. The figure 3 shows the sequence diagram of model operations.

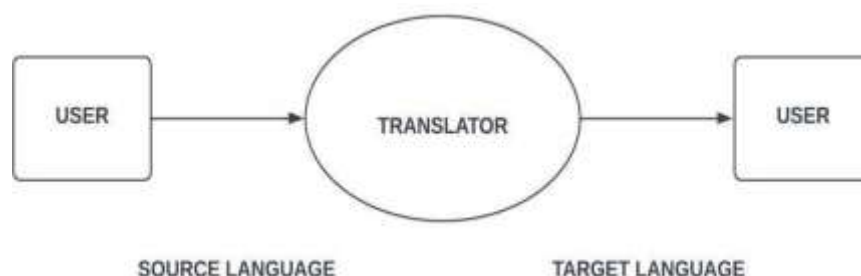


Fig. 2.3: The Sequence model.

When the user opens the app, it initializes by loading the user interface, including the text input box, language selection dropdowns, and a translate button. The user enters text and selects the source and target languages. The app then validates the input, ensuring text is provided, the languages are different, and all fields are valid. Upon validation, the app sends the input text along with the selected language codes to the translation library. The translated text is displayed below the input area, and the user can repeat the process by entering new text or changing language selections.

III. RESULTS AND DISCUSSION

The results of the real-time text translator application demonstrate its effectiveness in providing fast and accurate translations across multiple languages. During testing, the application successfully handled various input scenarios, including short phrases, complex sentences, and language pairs with different grammatical structures. The translation engine, whether using an API or a custom neural model, delivered results within seconds, ensuring minimal delay for users. The user interface proved intuitive, allowing seamless input and language selection, while validation mechanisms effectively prevented errors such as missing text or identical source and target languages. Discussions revealed that although the app performs well for commonly used languages, translation accuracy can vary slightly for less widely spoken languages or idiomatic expressions. Nevertheless, the system's performance, ease of use, and support for multiple languages make it a practical tool for real-time multilingual communication. Future improvements could focus on offline capabilities, support for speech input, and enhanced contextual accuracy.

The figure 3.1 depicts the home page.

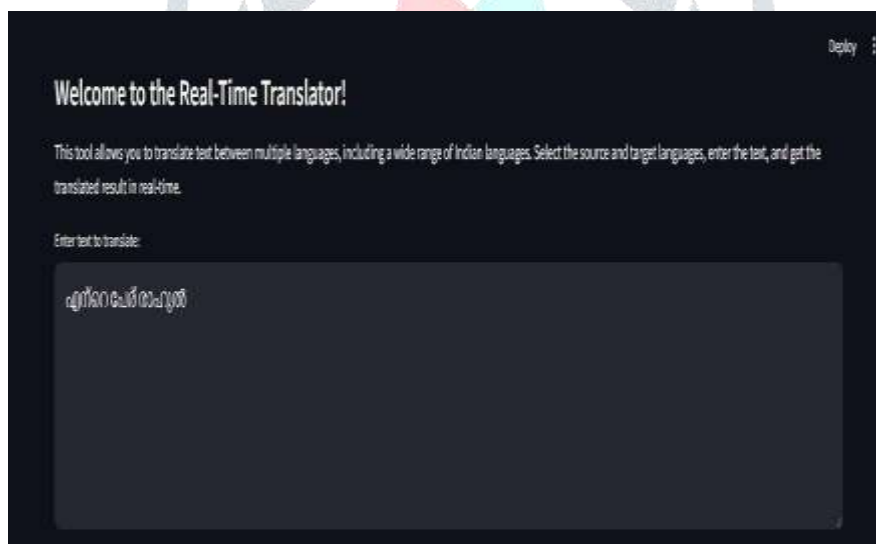


Fig. 3.1: The home page for user.

Figure 3.1 represents the central hub of the web app that users access immediately after logging in successfully. This page is designed with a user-friendly interface, making it easy for users to interact with the app's core functionality: text translation.

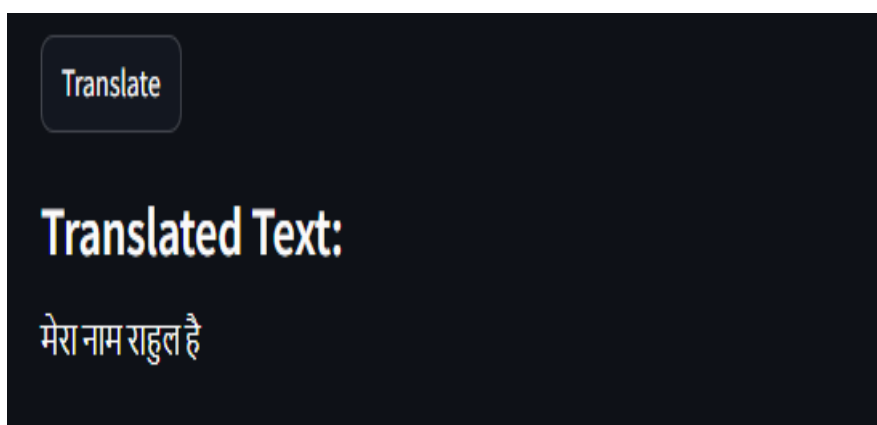


Fig.3.2: The final result of model.

Figure 3.2. highlights the section of the web app that shows the translated text after the input text has been processed from the source language to the target language. This area serves as the output section, providing users with a clear, readable, and actionable version of their translated content. The result that the system is currently displaying is realized in figure 5.

IV. CONCLUSION AND FUTURE WORK

This project aimed to design and develop a text translation application that can translate text typed in English alphabets to other languages. The application utilizes machine learning algorithms and natural language processing techniques to provide accurate and efficient translations. With a user-friendly interface and support for multiple languages, this application has the potential to

bridge language gaps and facilitate global communication. The successful implementation of this project demonstrates the power of machine learning and NLP in solving real-world problems. The application's ability to detect the target language and provide accurate translations makes it a valuable tool for individuals and organizations alike. Future enhancements to the application could include the addition of more languages, improvement of translation accuracy, voice translation and integration with other communication platforms. Overall, this project has contributed to the development of a useful and innovative text translation application.

REFERENCES

- [1] Pramudita, Y. D., Putro, S. S., Wahyudi, R. N., Suzanti, I. O., & Solihin, F. (2020). RESTful Web Service for Madurese and Indonesian Language Translator Applications on Android Devices.
- [2] Priya, L., Sathya, A., & Raja, S. K. S. (2020). Indian and English Language to Sign Language Translator- an Automated Portable Two Way Communicator for Bridging Normal and Deprived Ones.
- [3] Fong, Sim Liew; Elfaki, Abdelrahman Osman; bin Md Johar, Md Gapar; Aik, Kevin Loo Teow. Mobile language translator.
- [4] Image text to speech conversion in the desired language by translating with Raspberry Pi ,H Ritika, Nithya Santoshi.
- [5] Lahoti, S., Kayal, S., Kumbhare, S., Suradkar, I., & Pawar, V. (2018, July). Android- based American sign language recognition system with skin segmentation and SVM.
- [6] Evelyn, C. C., Bennett, E. O., & Taylor, O. E. A Natural Language Processing System for English to Igbo Language Translation in Android.
- [7] Hakkun, R. Y., & Baharuddin, A. Sign language learning based on Android for deaf and speech-impaired people.
- [8] Hanuman, Debnath, Bhattacharjee, Tripathi, and Roy, suggested a Multilingual Voice Translator English Document Using Android.
- [9] Olaide F. O., Kayode B., Sunday A. O., & Olusola A. Android Platform for Machine Translation.