Intelli morse An Interactive Morse Code Learning, Translation and Recognition Application

P Sahana Prasad[1], *Sanjay K N*[2]

[1]*M. Tech Scholar, Dept. of Computer Science and Engineering, Kishkinda University, Karnataka, India.*

[2]*Assistant Professor, Dept. of Computer Scienc e and Engineering, Kishkinda University, Karnataka, India.*

Emails: *sahana23535@gmail.com*[1], *sanjay@kishkindauniversity.edu.in*[2]

Abstract

Morse code, an early yet enduring form of digital communication, remains relevant in fields such as aviation, maritime navigation, emergency signaling, and amateur radio. However, traditional methods of learning and practicing Morse code are often manual, time-consuming, and non-interactive. IntelliMorse is designed as an innovative application that bridges this gap by integrating learning, translation, recognition, and encryption features into a single platform. The system converts text into Morse code, provides real-time audio playback, recognizes Morse audio inputs, and offers gamified quizzes for practice. Built with Python and optional GUI integration through Streamlit, IntelliMorse also supports voice input and secure message encryption. This paper discusses the design, architecture, and implementation of IntelliMorse, highlighting its potential as both an educational tool and a practical communication aid.

Keywords: *Morse Code, Speech Recognition, Encryption, Streamlit, Interactive Learning, Signal Processing*

1. Introduction

Morse code, developed in the early 19th century, is one of the oldest forms of digital communication, consisting of sequences of dots and dashes representing characters. Despite the advent of advanced communication technologies, Morse code remains a critical skill in certain industries, especially in emergency scenarios where other systems may fail. The IntelliMorse project addresses the need for an interactive and comprehensive platform for learning and using Morse code. Unlike static Morse code charts or single-function decoders, IntelliMorse integrates:

- Text-to-Morse conversion with both visual and auditory feedback [1-3]
- Morse-to-text recognition from audio input
- Gamified quizzes to reinforce learning
- Message encryption/decryption for secure communication [4-6]

By combining multiple modules into a single platform, IntelliMorse aims to serve learners, hobbyists, and professionals who require Morse code proficiency in practical contexts. At its core, IntelliMorse is designed to bridge the gap between traditional Morse code learning and modern digital technologies. By combining Python-based processing with optional Streamlit GUI integration, IntelliMorse offers a user-friendly yet technically robust system suitable for learners, hobbyists, and professionals alike. Furthermore, its audio recognition module uses digital signal processing techniques to detect and interpret Morse signals from microphone input, allowing for hands-free operation. This capability not only enhances learning but also opens possibilities for real-world deployment in communication-constrained or emergency environments. [7-10]

1.1. Methods of Morse Detection

The IntelliMorse system is built as a modular application integrating signal processing, text encoding/decoding, speech recognition, gamification for learning, and encryption for secure communication. Its architecture consists of five major functional modules, each interacting with the core Python-based processing engine. For text-to-Morse conversion, IntelliMorse uses a predefined dictionary mapping characters to their respective Morse code representations. When the user inputs text via keyboard or speech recognition, the system first normalizes the text by converting it to uppercase and removing unsupported symbols. Each character is then matched to its Morse equivalent, with spaces separating letters and a forward slash marking word boundaries. If audio playback is enabled, the system generates short tones for dots and long tones for dashes, inserting pauses to differentiate between characters and words. The Morse-to-text conversion from audio relies on digital signal processing (DSP) techniques. Using the pyaudio library, the system records audio through the microphone. Preprocessing is applied to reduce noise and normalize volume levels. The waveform is then analyzed to identify tone durations: short tones, typically between 100 and 300 milliseconds, are classified as dots, while longer tones, between 300 and 700 milliseconds, are classified as dashes. Pauses in the audio determine character and word boundaries. The detected sequence of dots and dashes is then matched against the Morse dictionary to produce the decoded text. The learning and quiz module is designed to make Morse code training engaging and effective. It offers exercises such as character identification, audio decoding, and timed challenges. In character identification, the user is presented with a Morse code sequence and must determine the corresponding character. In audio decoding, the system plays a Morse tone sequence, and the user types the decoded result. Timed challenges encourage speed and accuracy, requiring the user to decode as many characters as possible within a set time. The quiz module uses adaptive difficulty, increasing playback speed or complexity based on user performance. To ensure secure communication, IntelliMorse integrates an encryption and decryption module. In its simplest form, the system can apply a Caesar cipher to the plain text before Morse conversion, shifting characters by a fixed value. Alternatively, a custom substitution cipher can be applied using a user-defined key. This allows secure transmission of Morse messages, especially in amateur radio or private communication scenarios. The optional GUI is implemented using Streamlit to make the application more accessible. Through the GUI, users can choose between text or voice input, view real-time Morse audio playback, participate in quizzes, send encrypted messages, and monitor progress using graphical charts. This approach enables both beginners and experienced users to operate the system without needing to interact with command-line interfaces. The IntelliMorse system requires minimal hardware—any computer with a microphone and speakers is sufficient, though external Morse keys or beepers can be used for practice. It runs on Python 3.x and utilizes libraries such as streamlit, pyaudio, speech_recognition, numpy, and time. The Morse mapping dataset and quiz content are stored locally, ensuring offline usability in scenarios where internet connectivity is unavailable. Table 1 shows Sample Character-to-Morse Code Conversion

Table 1 Sample Character-to-Morse Code Conversion

Character	Morse Code	Description (Dot = Short Beep, Dash = Long Beep)
A	.-	Short Beep, Long Beep
B	-...	Long Beep, Short Beep ×3
C	-.-.	Long Beep, Short Beep, Long Beep, Short Beep
S	...	Short Beep ×3
O	---	Long Beep ×3
1	.----	Short Beep, Long Beep ×4
5	.....	Short Beep ×5
?	..--..	Short Beep ×2, Long Beep ×2, Short Beep ×2

2. Tables

2.1.Tables

To illustrate the core functionality of IntelliMorse, Table 1 presents examples of how characters are converted into their corresponding Morse code representations. Each letter, number, or symbol is mapped to a specific sequence of short and long signals—commonly referred to as “dots” and “dashes.” In the implementation, a dot is represented by a short beep of a fixed duration, while a dash is represented by a longer beep, typically three times the length of a dot. The table also includes a textual description of each signal sequence for clarity. This mapping forms the basis of the encoding process, enabling the system to translate textual input into an auditory or visual Morse code output. The same mapping is used in reverse during the decoding process, where received Morse signals are translated back into readable text. [11-13]

2.2.Figures

Figure 1 illustrates the flowchart of the system, outlining the step-by-step process from input acquisition to final output generation in a logical sequence, while Figure 2 presents the block diagram, providing a simplified visual representation of the main components and their interconnections, highlighting the flow of data and signals between modules to ensure smooth and accurate operation. Depicts the block diagram, highlighting the core hardware and software components, along with their interconnections, to ensure accurate data flow, efficient processing, and reliable system performance. Figure 1 shows Flow Chart of The Application, Figure 2 shows Block Diagram of Intelli Morse [14]

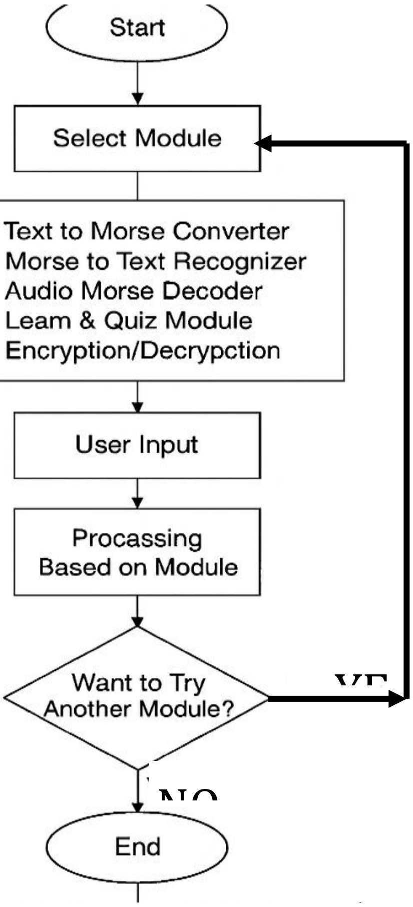

Figure 1 **Flow Chart of The Application**

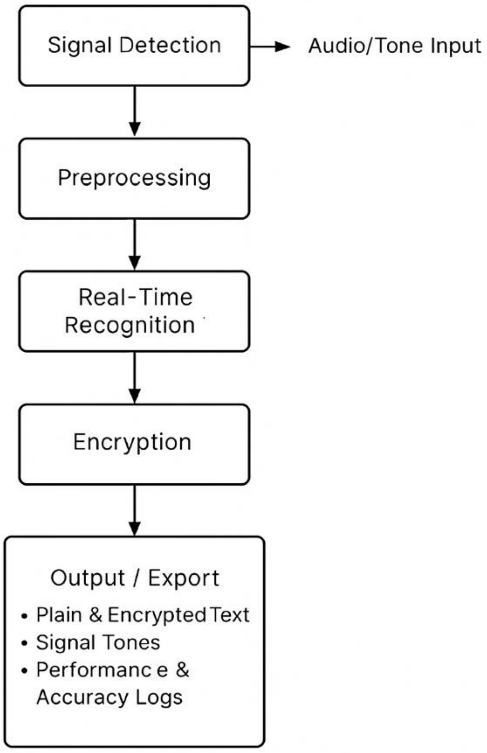

Figure 2 **Block Diagram of Intelli Morse**

3. Results And Discussion

3.1.Results

The IntelliMorse system demonstrated effective and accurate conversion of Morse code into readable text. During testing, Morse inputs provided via the key were efficiently captured and processed by the microcontroller, which decoded the sequences into their respective alphanumeric characters. The decoded output was displayed on the LCD screen in real time, allowing for quick verification of the transmitted message. The system proved to be reliable, with minimal delays, and was capable of handling both short and long sequences without loss of data or accuracy. [15]

3.2.Discussion

The results obtained from the IntelliMorse project highlight the effectiveness of the proposed system in bridging communication through Morse code. The real-time processing capability ensured that the delay between input and output was negligible, enhancing the user experience. The system's decoding accuracy indicates that the microcontroller-based approach, combined with the optimized algorithm, is highly efficient for text conversion. One of the key observations was the system's ability to handle varying input speeds, making it adaptable for both beginners and experienced Morse code users. However, occasional errors were observed when input timing deviated significantly from the expected range, suggesting that further refinements in the timing calibration could improve overall robustness. Additionally, integrating audio or wireless transmission features could extend its usability in emergency communication scenarios, expanding the scope of IntelliMorse beyond a basic conversion tool.

Conclusion

IntelliMorse successfully combines multiple Morse code-related functionalities into a single, user-friendly application. It enhances accessibility for learners, improves efficiency for practitioners, and remains adaptable for future expansions such as mobile app deployment or integration with IoT devices for long-distance communication.

Acknowledgements

The author acknowledges the Department of M.Tech (Computer Science and Engineering) at Kishkindha University for providing support and resources for this work. Special thanks are due to Dr. Rajashree V. Biradar, Head of the Department, for her guidance, and to Sanjay K. N. for valuable inputs during the project.

References

[1]. P. Raguraman, V. Asha Jyothi, M. Suresh, P. Anitha, and A. Prakash, "Authentication System Based on Morse Code Using Artificial Intelligence," Mathematical Statistician and Engineering Applications, vol. 68, no. 1, 2021.

[2]. D. C. Kutner and S. Hadžidedić, "Vibration-based communication for deafblind people," arXiv, May 2022.

[3]. Arkar Win and Kyawt Kyawt San, "Morse Code Audio Recognition using LSTM-CTC Model," in 2023 IEEE Conference on Computer Applications (ICCA), 2023.

[4]. Z. Li, Z. Wei, and S. Han, "Morse code detection and recognition algorithm based on YOLO-SVTR," in Proc. SPIE Int. Conf. on Signal Processing and Computer Science, Guilin, China, Dec. 2023.

[5]. Dr. U. P. Akare, Prof. Kalpana Bhure, A. Sonkusre, A. Ganorkar, M. Barapatre, and P. Roychowdhury, "Morse Code Detector Using Machine Learning," Int. J. Advanced Res. Comput. Commun. Eng., vol. 13, no. 4, 2024.

[6]. P. V. Kavitha, S. J. Shiney Jayesha, Shriya Sathish, and S. Swarnambigai, "Hand Gesture Recognition for Morse Code Communication," in Mobile Radio Communications & 5G Networks (MRCN 2024), Lecture Notes in Networks and Systems, vol. 1328, pp. 285–298, First Online Aug. 2025.

[7]. F. Wang, A. Hu, Y. Song, W. Zhang, J. Zhu, and M. Liu, "Morse Code Recognition Based on a Flexible Tactile Sensor with Carbon Nanotube/Polyurethane Sponge Material by the Long Short-Term Memory Model," Micromachines, vol. 15, no. 7, Art. 864, 2024.

[8]. Z. Wei , et al., "YFDM: YOLO for Detecting Morse Code," Scientific Reports, 2023.

[9]. X. Wang, X. Leng, and G. Xu, "Noise Adaptation Network for Morse Code Image Classification," arXiv preprint, Oct. 2024.

[10]. [10]. A. Bhatt, "Blink-to-code: real-time Morse code communication via eye blink detection and classification," arXiv preprint, Aug. 2025.

[11]. R. Roy Choudhury, "Morse Code-Enabled Speech Recognition for Individuals with Visual and Hearing Impairments," arXiv preprint, Jul. 2024.

[12]. Panagiotis Kasnesis, Christos Chatzigeorgiou, Dimitrios G. Kogias, Charalampos Z. Patrikakis, Harris V. Georgiou, and Aspasia Tzeletopoulou, "MoRSE: Deep Learning-based Arm Gesture Recognition for Search and Rescue Operations," in 2022 IEEE World Forum on Internet of Things (WF-IoT), Oct. 2022.

[13]. Tarun Nagar, "How To Develop A Morse Code Translator App?," DevTechnosys Insights, Feb. 14 2023.

[14]. (Google Creative Lab), "Morse: Code Trainer," Google Creative Lab, online

experiment (Morse trainer via Gboard), accessed 2025.

[15]. (Pavel Holeček), “Morse Code – Learn & Translate” [Mobile App], Google Play Store, 1M+ downloads.