Morse Code Detector

Dr.U.P. Akare[1], Prof. Kalpana Bhure[2], Adarsh Sonkusre[3], Atharva Ganorkar[4], Mayank Barapatre[5], Pratyush Roychowdhury[6]

1-2Assistant professor, Department of Artificial Intelligence and Data Science, Priyadarshini College of Engineering, Nagpur

3-7 Department of Artificial Intelligence and Data Science, Priyadarshini College of Engineering, Nagpur

Abstract— **The aim of this research project is to create a Morse code detecting system by utilizing the ESP32 microcontroller's capabilities and machine learning capacity. An enduring communication technique, Morse code can be used for emergency signals and low-power communication among other things. In order to develop a flexible and effective Morse code detector, we plan to integrate contemporary technology with conventional communication methods. The main goal of the project is to create a small, inexpensive device that can precisely recognize and decode inputs that contain Morse code messages. The core processing unit, the ESP32 microcontroller, is responsible for preprocessing and signal acquisition. It also offers smooth networking choices for remote control and data transfer. In order to recognize Morse code signals, we use machine learning methods.**

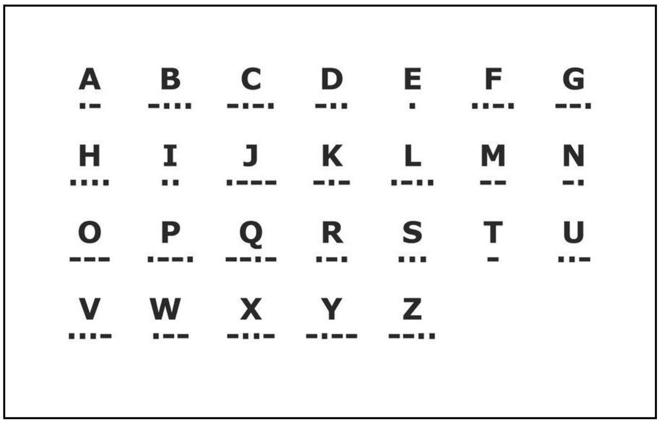
Keywords—ESP32, Convolutional neural networks, Recurrent neural networks, Signal preprocessing, Audio capture, Real-time recognition, internet of Things (IoT), Remote monitoring, Communication technology, Emergency signaling, Low-power communication.

I. INTRODUCTION

Communication From smoke signals to telegraphs, telephones to the internet, communication has changed dramatically over time. Using a series of dots and dashes to encode data, Morse code is a straightforward but efficient method of long-distance communication that is among the oldest and most reliable. Even if Morse code seems outdated in the current digital era, it is still useful for amateur radio, emergency signaling, and even as a hobby.

The Morse code was created in the 1830s by Samuel Finley Breese Morse (1791–1872). The code employed by American inventor and artist Morse with his electrical telegraph system was created. The alphabetic letters in the code are represented by a succession of dots and dashes An assembly of European countries established in 1851 To accommodate letters containing diacritical markings, a meeting of European nations developed the International Morse Code in 1851. About 1844, the Morse telegraph system was put into operation. When the system sensed the presence of electric currents, it made impressions on a paper tape. KFS at Half Moon Bay was the final American ship-to-shore station to send a message in Morse code; it did it on July 12, 1999. The age of the maritime radio-telegrapher came to an end as satellite technology took the place of Morse code.

While the numerals (0–9) follow a pattern and are easy to learn, the majority of Morse code symbols must be memorized by memory. There are five characters in all that comprise each number.



In the current technological era, our goal is to connect the dots between digital world and Morse code. Our project, "Morse Code Detection Using Machine Learning and ESP32," uses the adaptable ESP32 microcontroller and state-of-the-art machine learning techniques to build a system that can decode real-time Morse code communications. This creative idea showcases the power of automation and integration by fusing cutting-edge technology with conventional communication techniques.

A. *Current trends of morse code*

Amateur radio communication: Prior to the general availability of voice-capable radio transmitters around 1920, amateur radio operators only utilized morse code. It was a normal means of communication used by interpreters, the military, and other groups.

The International Telecommunication Union required knowledge in Morse code as a prerequisite for amateur radio licenses globally until 2003. But in recent years, many nations have phased out this need. These days, you may get a ham radio license in the majority of countries without knowing Morse code. Several nations, including the US, UK, Australia, and NZ, are home to the International Society for the Preservation of Morse Code. Aviation and military communication: Pilots, both military and private, were needed in the 1930s to employ Morse code. Both professional and amateur aviation, naval, and geographic systems employ the International Morse Code.

In aircraft, audio navigation assistance identification is done using Morse code. When you listen in to a VOR, NDBs and VORs broadcast their identification letters as beeps audible on the audio channel. Pilots must recognize aircraft call signs and comprehend Morse code, according to FAA regulations. Morse code is used in the military to send signals across great distances. Teams from Special Forces, for instance, may communicate across long distances using Morse code.

Signaling an emergency: Three dots, three dashes, and three dots is the Morse code SOS distress signal. It is utilized to make instantaneous emergency aid requests in circumstances like shipwrecks, plane collisions as well as other crises. SOS is represented by the letters "SOS" (... ---...). in Morse code. Three dots, three dashes, and then three more dots are sent, with no spaces between the letters, to be communicated. Sailors most likely came up with the phrase "Save Our Ship" to indicate to other boats in need of assistance. You can practice banging on objects or flipping light switches to transmit an SOS The Automatic Transmitter Identification System (ATIS) uses Morse code to identify uplink sources of analog satellite transmissions.

Here are some details about ATIS: Transmission: ATIS is a Morse code transmission sent on a subcarrier of 7.1Hz Protocol: The ATIS signal should be International Morse Code keyed by a 1200 Hz ± 800 Hz tone Repetition: The ATIS signal should repeat every 25-30 seconds. Activation: The ATIS signal must activate automatically any time the station is transmitting. Morse code uses short and long signals, called dots and dashes, to represent letters and numbers. Messages are sent by tapping, drumming, buzzing, clicking, or flashing light. Morse code is considered a more reliable communication method than more modern means since its signals need much less broadcasting power than voice messages.

II. Objective

A. Real-time Morse Code Detection

Our main objective is to create a system that can instantly recognize and interpret communications encoded using Morse code. This is recording Morse code-containing audio signals, processing them, and turning them into legible text. The goal of the Real-time Morse Code Detection project is to create a system that uses analog inputs to effectively detect and interpret Morse code signals in real-time. In communication systems throughout history, Morse code a technique for encoding text letters using a series of dots and dashes was employed. This project makes use of contemporary technology to automatically decipher Morse code messages, opening up a range of uses such as educational or mobility-impaired people's communication.

Create and put into operation a real-time Morse code audio signal processing system. identification. To improve accuracy, investigate machine learning algorithms for interpreting Morse code. Create an intuitive user interface so that users may communicate with the Morse code detection system in real time. Assess how well the system performs in various noise levels and situations.

B. Using Machine Learning to Interpret Morse Code

We will use machine learning methods to extract Morse code patterns from audio sources in order to accomplish real-time detection. To enable accurate and efficient interpretation, a model will be trained on a collection of recordings in Morse code. The goal of the "Machine Learning for Morse Code Interpretation" project is to improve the precision and effectiveness of Morse code interpretation from audio signals by utilizing cutting-edge machine learning techniques. Due to differences in signal properties, real-time interpretation of the ancient communication method known as Morse code is difficult. This endeavor investigates the use of machine learning techniques to strengthen the interpretability of Morse code, allowing for more precise and dependable communication.

Apply machine learning techniques to audio signals to identify Morse code patterns. Teach models to distinguish between Morse code pauses, dashes, and dots. Make the system more flexible in response to different signal intensities and noise levels in the background Assess how well machine learning models comprehend Morse code in real time.

C. Integration of ESP32 Microcontroller

The ESP32 microcontroller is renowned for its adaptability and range of connectivity choices. Its capabilities will be put to use in order to process data, record audio signals, and display decoded Morse code messages on a variety of output devices, like serial monitors or LED displays. The ESP32 microcontroller is being integrated into a variety of systems and applications as part of the "ESP32 Microcontroller Integration" project.

The ESP32, renowned for its adaptability and power, is used in a variety of applications to improve connectivity, functionality, and performance. An overview of the ESP32 microcontroller's integration process, applications, difficulties, and results is given in this article Use the ESP32 microcontroller in a variety of applications and projects. Take advantage of the Bluetooth and Wi-Fi features for improved connectivity. Examine the ESP32's possibilities in the Internet of Things. Examine the ESP32's potential for Internet of Things applications. Assess the ESP32's functionality, dependability, and adaptability in a range of situations.

D. Easily Expandable and Customizable

We'll make sure our system can be readily expanded and customized. Additional functions like translating text into other languages, generating Morse code, and integrating with external communication devices will be available for users to incorporate. Put in place a Morse code detecting system based on machine learning. Provide for system expansion in order to incorporate more features and functionalities. Provide alternatives for user preferences and particular application needs to be customized. Assess the system's responsiveness to different circumstances and real-time detection performance.

III. Literature Review

A. Historical Significance of Morse Code

The original idea behind Morse code was to encode textual data into a series of dots and dashes that would stand in for letters and numbers. With the widespread use of telegraphy, its adoption surged, transforming long-distance communication in the late 19th and early 20th centuries. It was a mainstay in early radio broadcasts, military operations, and maritime communication due to its effectiveness, versatility, and simplicity.

Although Morse code is no longer useful for general communication due to technological improvements, it is nevertheless used in specialized applications including amateur radio, emergency signals, and instructional settings. The underlying properties of Morse code, which express information in binary, have sparked ongoing research into its potential uses in the digital era.

B. Modern Technology and Morse Code

The fact that Morse code is still used today despite its age is demonstrated by the numerous applications and technologies that include it. With the use of switches, visual signals, or sound, among other input techniques, people with impairments can now communicate thanks to the use of Morse code in assistive technology. It is also appropriate for situations where bandwidth is limited or communication must be reliable under difficult circumstances because to its robustness and simplicity.

C. Detectors of Morse Code

Morse code detectors are instruments that automatically decipher Morse code signals, enabling accurate and efficient communication, thanks to technological advancements. These detectors identify dots, dashes, and spaces in the Morse code sequences using a combination of machine learning and acoustic signal processing algorithms.

D. Methods of Signal Processing

Signal processing for audio is essential part of detectors for Morse code. Incoming audio signals are filtered and analyzed using a variety of techniques to separate background noise from Morse code patterns. Accurate interpretation is made possible by techniques like wavelet transformations and Fourier analysis, which help to identify pertinent aspects from the signal

E. Methods for Machine Learning

The capabilities of Morse code detectors have been greatly improved by the incorporation of machine learning. Models for Morse code interpretation is trained using supervised learning approaches such as decision trees, neural networks, and support vector machines. These models pick up on patterns and adjust to varying signal intensities, speeds, and surroundings.

F. Problems and Future Directions

Despite the resiliency and adaptability of both Morse code and Morse code detectors, there are still issues. Accurate detection is constantly hampered by background noise, signal distortions, and changes in signal properties. In order to address these issues, future research may concentrate on improving machine learning models and maybe adding adaptive algorithms that can adapt dynamically to changing circumstances.

Furthermore, new research directions are made possible by the incorporation of Morse code with cutting-edge technologies like the Internet of Things (IoT). Morse code emphasizes versatility and ongoing relevance in a variety of technology landscapes by serving as a lightweight and energy-efficient communication protocol in Internet of Things devices.

In conclusion, despite its lengthy history, Morse code still has an impact on contemporary communication systems. Propelled by developments in signal processing and machine learning, as an example of how flexible this age-old mode of communication can be. Because of its adaptability and simplicity, Morse code can be used in a variety of ways as technology develops, leading to new advancements in communication systems.

IV. Methodology/Planning of Work

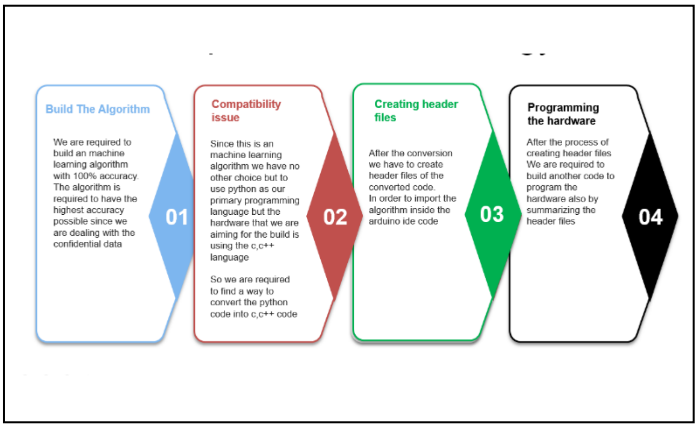


Fig.1

A. Create the Algorithm

Our task is to create a machine learning algorithm that is as accurate as possible. We need the algorithm to be as accurate as possible because we are working with sensitive data. Developing a machine learning system that can attain optimum accuracy is a difficult task, particularly in practical situations. A number of issues, including as data noise, model complexity, and restrictions in the algorithm's underlying assumptions, could make achieving perfect accuracy impracticable. While aiming for high accuracy is praiseworthy, in order to guarantee the algorithm's efficacy in a variety of scenarios, it is imperative to preserve a balance between model complexity and generalization. The process of developing algorithms should incorporate thorough testing, validation, and optimization in order to tackle this. Methods like cross-validation and hyperparameter tuning, Careful and meticulous data preprocessing can help to improve accuracy. It's important to recognize, nevertheless, that there may be circumstances in which obtaining optimal accuracy is not possible and may call for trade-offs. Furthermore, security needs to be the first focus while working with sensitive data. Creating a secure machine learning algorithm requires the use of encryption methods, access controls, and making sure that data protection laws are being followed.

B. Compatibility issue

Because this is a machine learning algorithm, we are forced to use Python as our main programming language, even if the hardware we are trying to develop is written in C and C++. Finding a means to translate the Python code into C or C++ is therefore necessary.

The incompatibility occurs when the target hardware utilizes C or C++ and the machine learning algorithm's primary programming language is Python. Python code can be translated into C or C++ code using a variety of tools and libraries to close this gap. Using libraries like Pybind11, micromelgen or tools like Cython can help with this conversion process. Writing code is simple with the Cython programming language.

C. Creating header files

After the conversion we have to create header files of the converted code. In order to import the algorithm inside the

Arduino ide code. Once the Python code is successfully converted into C or C++ code, the next step is to create header files. Header files (.h) serve as interfaces between different parts of a program. They contain declarations and macro definitions that are essential for incorporating external code into a project. Creating header files of the converted code is necessary for importing the machine learning algorithm into the Arduino Integrated Development Environment (IDE).The header files encapsulate the essential functions, variables, and structures needed for the algorithm to function within the Arduino IDE. This step ensures modularity and ease of integration, allowing the algorithm to be seamlessly included in Arduino projects.

D. Programming the hardware

After the header file creation procedure, the hardware is programmed. To program the hardware, we must write additional code that summarizes the header files. The last step is to program the hardware—in this case, using the Arduino IDE—after the header files have been produced. An interface between the hardware and the machine learning algorithm is provided by a new code that summarizes the functionality of the header files. Memory limits and processor power should be taken into consideration while optimizing this code for the target hardware specs.

Microcontroller programming is made easier with the Arduino IDE, and the compiled code needs to be optimized to run quickly on the particular hardware. This procedure turns the machine learning algorithm into an essential component of the hardware, enabling it to carry out operations according to the predictions or classifications made by the algorithm.

In conclusion, thorough consideration of algorithmic complexities and security precautions is necessary to create a machine learning algorithm with maximum accuracy. The successful integration of the method into a secure data processing environment requires addressing Python and C/C++ compatibility difficulties, generating header files, and effectively programming the hardware. It takes skill in both machine learning and security systems development to strike a careful balance between achieving high accuracy and protecting data confidentiality.

V. BLOCK DIAGRAM

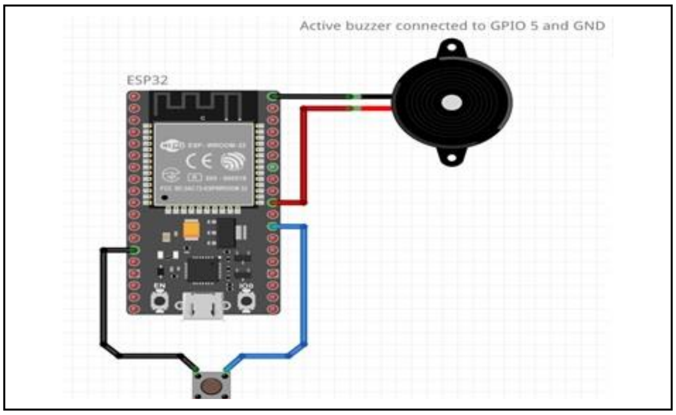


Fig.2

A. Components

ESP32 Microcontroller: Combining Bluetooth and Wi-Fi functionality, the ESP32 is a multipurpose microcontroller. It is frequently utilized in many IoT initiatives and applications.

The ESP32's GPIO (General Purpose Input/Output): pins are used to communicate with external hardware, including switches, actuators, and sensors.

Buzzer: When an electrical signal is given to an electronic component, the component emits sound.

The buzzer in this circuit is wired to the ESP32's GND and GPIO5 pins.

Tap Switch (Tactile Switch): A tap switch, sometimes referred to as a push-button, is a momentary switch that must be pressed in order for a connection to be made.

The ESP32's GND and GPIO16 pins are linked to the tactile switch.

VI. EXPECTED OUTCOMES

The "Morse Code Detection Using Machine Learning and ESP32" project is anticipated to yield a number of technological advances, usability enhancements, and possible uses. The creation of a working system with real-time Morse code message detection and decoding capabilities is the main result. This implies that the system must be able to decipher transmitted Morse code signals and translate them into legible text. The following are the expected results:

A. Machine Learning Model

The goal of the project is to develop and hone a machine learning model intended especially for the interpretation of Morse code. In order to ensure dependable and effective decoding, this model should demonstrate high accuracy in identifying Morse code patterns from audio inputs.

B. ESP32 Integration

Making use of the ESP32 microcontroller's capabilities, the project should be able to process data, record audio signals, and output decoded Morse code messages on a variety of output devices (such as LED displays or serial monitors). This integration ought to show off the ESP32's flexibility and adaptability in communication systems.

C. User-Friendly Interface

The system can be configured to have an intuitive user interface that makes it possible for users to engage with it without any technical knowledge at all. It should be simple and obvious for users to enter messages in Morse code for decoding and to receive the decoded messages.

D. Expandability and customization

Future expansion and customization should be supported by the project's design. It should be possible for users to incorporate extra functions with ease, such translating text into other languages, creating Morse codes, and interacting with external communication devices. The project's adaptability and long-term utility are contingent upon its extensibility.

E. Tutorials and Documentation

The project should come with thorough tutorials and documentation. Users should be able to replicate the system and modify it to suit their own requirements with the help of these resources. It is ensured that others can build upon and profit from the project's efforts via clear and comprehensive documentation.

F. Demonstration of Morse Code's Relevance

The project should demonstrate the continued relevance of Morse code in the contemporary digital environment, in addition to its technical results. The project demonstrates how modern technology may be combined with Morse code, which serves to highlight the need of simplicity, dependability, and flexibility in communication systems.

REFERENCES

[1] A. Win and K. K. San, "Morse Code Audio Recognition using LSTM-CTC Model," 2023 IEEE Conference on Computer Applications (ICCA), Yangon, Myanmar, 2023, pp. 393-398, doi: 10.1109/ICCA51723.2023.10181830. keywords: {Deep learning;Codes;Hidden Markov models;Computer applications;Media;Feature extraction;Classification algorithms;Morse code;MFCC;LSTM;CTC;Audio recognition;Feature extraction},

[2] S. M, N. Kolkar, S. G. S and K. Kulkarni, "Morse Code Detector and Decoder using Eye Blinks," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2021, pp. 1-6, doi: 10.1109/ICIRCA51532.2021.9545039. keywords: {Codes;Costs;Databases;Software algorithms;Ear;Detectors;Software;Eye blinks;Morse code detector;Eye Aspect Ratio;OpenCV},

[3] P. Nalajala, B. Godavarth, M. L. Raviteja and D. Simhadri, "Morse code generator using Microcontroller with alphanumeric keypad," 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, India, 2016, pp. 762-766, doi: 10.1109/ICEEOT.2016.7754788. keywords: {Microcontrollers;Liquid crystal displays;Light emitting diodes;Aerospace electronics;Generators;Transmitters;Keyboards;International Morse Code;8∗4 Matrix keypad;Radio telegraphy;Binary Format;8051 Microcontroller;LED and LCD Display},

[4] W. Li and K. Wang, "Research on Automatic Decoding of Morse Code Based on Deep Learning," 2019 International Conference on Intelligent Computing, Automation and Systems (ICICAS), Chongqing, China, 2019, pp. 488-491, doi: 10.1109/ICICAS48597.2019.00107. keywords: {Morse code, automatic decoding, deep learning, frequency drift, Manual deviation},

[5] M. Babiuch, P. Foltýnek and P. Smutný, "Using the ESP32 Microcontroller for Data Processing," 2019 20th International Carpathian Control Conference (ICCC), Krakow-Wieliczka, Poland, 2019, pp. 1-6, doi: 10.1109/CarpathianCC.2019.8765944. keywords: {Microcontrollers;Microprogramming;Monitoring;Python;Embedded systems;Wireless fidelity;ESP32;IoT;microcontroller;measurement;display;development boards;environment},

[6] T. R. N and R. Gupta, "A Survey on Machine Learning Approaches and Its Techniques:," 2020 IEEE International Students' Conference on Electrical,Electronics and Computer Science (SCEECS), Bhopal, India, 2020, pp. 1-6, doi: 10.1109/SCEECS48394.2020.190. keywords: {Machine learning algorithms;Supervised learning;Machine learning;Reinforcement learning;Semisupervised learning;Predictive models;Unsupervised learning;Machine learning;Supervised learning;Unsupervised learning;Semi-supervised learning;Reinforcement Learning.},