ISSN 2063-5346

GESTURE RECOGNITION USING TINYML

Jalaja S [1], Aishwarya K [2], Divya PS[3], Sripradha M[4], Pavithra M[5], Thapasya K[6]

Article History: Received: 01.02.2023 Revised: 07.03.2023 Accepted: 10.04.2023

Abstract

Gesture recognition using TinyML and Wio Terminal is a project where machine learning models have been used to recognise gestures in real-time. Use of machine learning and microcontrollers in this project has opened up opportunities for various applications in fields such as human-computer interaction, robotics, VR & AR gaming experience, Web3, spatial computing technology, and many others. A Convolutional Neural Network (CNN) model that has been trained to identify hand gestures from a dataset of real-world gestures serves as the foundation for this research. The dataset is processed to extract relevant features, and the model is trained using Codecraft, a popular tiny machine learning application. The Wio Terminal, a compact and versatile device, is used to capture live input from the user's hand gestures using its built-in accelerometer and gyroscope sensors. The detected gestures are classified in real-time using the deployed machine learning model and the corresponding actions are displayed on the Wio Terminal's LCD screen. The system is a workable option for numerous real-time gesture recognition applications due to its effective resource management and small footprint in wastage of resources. The project's outcomes show that real-time gesture detection applications employing TinyML are feasible, opening up a larger spectrum of possibilities. Compact, affordable, and low-power gesture recognition systems that may be employed in a variety of applications can now be built using machine learning and microcontrollers.The demonstration also illustrates the potential of machine learning models in embedded systems, which may pave the way for the future creation of more sophisticated systems and applications in nano technology using ML.

Overall, Gesture recognition using TinyML and Wio Terminal is a promising project that showcases the potential of machine learning and microcontrollers in developing practical solutions for real-time gesture recognition applications. This project results can be used for further research and development in Embedded ML and could lead to further advancement in live capturing of spatial technology..

Keywords - Gesture recognition, TinyML, Wio terminal, Machine learning, Spatial computing technology, Microcontrollers, Human-computer interaction, Robotics, VR & AR gaming, Web3, Convolutional neural network, Codecraft, Accelerometer, Gyroscope.

[4],[5],[6]Final Year UG Students, Department of Electronics and Communication Engineering,

[1], [2], [3] UG Professors, Department of Electronics and Communication Engineering,

Vel Tech High Tech Dr Rangarajan Dr Sakunthala Engineering College

Avadi- Vel Tech Road Chennai, India-600062

Corresponding author: msripradha@gmail.com

No 2B, Vishwamelu Towers, Nehru Nagar, Chromepet, Chennai-44

DOI:10.31838/ecb/2023.12.s1-B.228

I. INTRODUCTION

Mathematical algorithms are interpreted to extract human gestures in the process of gesture recognition [3]. Hand gestures are a common mode of communication, and gesture recognition technology aims to bridge communication gaps by translating sign language or enabling users to interact with devices without physically touching them. This technology has the potential to replace conventional input methods in everyday devices [6-10]. The Wio Terminal is a microcontroller-based device that integrates various sensors and interfaces, making it a versatile and compact device. The project combines this device with machine learning, a branch of artificial intelligence, to develop a gesture recognition system. The use of tinyML, which is embedded machine learning, allows the model to be deployed on microcontrollers, such as the Wio Terminal, that have limited resources, including a 32-bit CPU and a few kilobytes of memory. Microcontrollers are low-cost electronic components that consume minimal power and are widely used in modern consumer, medical, automotive, and industrial equipment. Billions of microcontrollers are sold each year, and there are likely hundreds of billions of them in use currently. However, these devices are not typically equipped with machine learning capabilities, making the combination of the Wio Terminal and tinyML a promising development for real-time gesture recognition applications. By combining machine learning and microcontrollers, the project demonstrates the potential for creating compact and low-cost gesture recognition systems that can be used in various applications. The integration of tinyML [1-2] with the Wio Terminal allows for efficient use of resources and real-time gesture recognition.

II. RELATED WORKS

1. The Paper 1 [1]

In this research paper, a general overview of Tiny Machine Learning (TinyML) is illustrated. Based on the fast edge computing system, hardware and software can be embedded. This reduces the dependency of third party cloud centres and lowers the latency of processing. 47 papers were collected and analysed to get a comprehensive understanding of the TinyML framework. Beginning in 2019, early TinyML publications were collected. Hardware, frameworks, datasets, use cases, algorithms, and models are five categories of TinyML literature that are looked at. Hence we get a holistic approach of solving hardware driven machine learning problems.

2.The Paper 1I [2]

Despite the fact that hand gesture detection has been thoroughly investigated using sensing modalities such as IMU, electromyography, and cameras, a small, power-efficient on-board inferencing solution is still difficult to come across. In this study, they present a wristband with capacitive sensing for on-board hand motion detection that is surrounded by four single-end electrodes. Seven hand movements of users with a wristband tied to their hands are identified accurately with precision of 96.4%. Later the sensor edge data is processed using a single convolutional hidden layer, for further results. This paper tells us the efficient usage of microcontroller based edge devices in the field of medicine.

3. The Paper 1II [3]

Compared to traditional interaction like using keyboard, mouse, pen, we get a more efficient hand interaction using image vision based interaction. This is increasingly proved to be more natural to use than others. In this study, they recommended a trustworthy live hand motion identification method. In this method, a specific hand gesture must be performed before the hand can be detected, tracked, and segmented using motion and colour cues. Since we usually have an aspect ratio problem, in order to enhance the experience, a scale space theory framework is used. The outcomes of these tests show that their technique functions effectively when utilised for picture browsing navigation. Although image recognition is used here, the latency and energy consumed is increased to a larger extent. We intend to bridge this gap with our project.

4. The Paper 1V [4]

As a nonverbal form of communication, hand gestures are used in a variety of settings, including home automation, spatial technology, human-computer interaction (HCI), robot control, and deaf-mute systems. In research articles that use hand gestures, several different techniques have been employed, including those relying on instrumented sensor technology and computer vision. For example, posture and gesture, dynamic and static, or a combination of the two, are all subcategories of hand signs. The research on hand gesture strategies is analysed in this paper, with the series of benefits and drawbacks of employing use cases in diverse settings.

5. The Paper V [5]

The Internet of Things has enabled optimal machine learning models with efficient and limited energy requirements. Because of the unpredictable environments with remote applications running on battery free TinyML devices is still difficult to deploy. In this paper, they demonstrated deploying tinyML algorithms and use case tasks on IoT devices without batteries while managing their energy use. They examine the trade-offs between various inference techniques, determining when it is preferable whether to send locally to the microcontroller or send the data to a huge external cloud service where a potent machine learning model is used while taking into account energy, accuracy, and time constraints. They have defined an energy consuming - aware decision model to determine which of the above mentioned two options is the superior option. We have used this paper to substantiate our paper on using TinyML for an energy-aware machine learning model.

III. TECHNICAL ARCHITECTURE

The technical architecture of Gesture recognition using TinyML and Wio Terminal involves the integration of various sensors to enable recognising hand gestures in real time [5]. The system is built around the Wio Terminal, which serves as the main interface for capturing sensor input and displaying the recognized gestures. The Wio Terminal includes various sensors, such as an accelerometer and a gyroscope, for capturing data related to hand movements. An embedded machine learning model is developed on hand gesture movements' dataset, which is used to process this sensor data. The model is deployed using the TinyML, enabling it to run on a microcontroller embedded in the Wio Terminal. The system uses an algorithm to classify the detected hand gestures in real-time and display the corresponding action on the LCD screen. With potential applications in areas including robotics, human-computer interaction, and assistive technology, the system's technical architecture overall offers a small and effective solution for real-time hand gesture identification employing sensors. Figure 1.1 below explains the flow diagram of the entire process from setting a goal to deployment of the application.

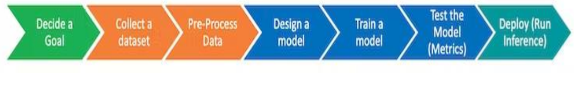

Figure 1.1: Process flow

IV. EXISTING METHOD

Gesture recognition has become an increasingly popular area of research in computer vision and machine learning. Two popular tools used in gesture recognition projects are OpenMV and OpenCV [4]. OpenMV is a low-power microcontroller designed specifically for computer vision applications. It features an onboard camera, and its firmware supports a range of algorithms for image processing and machine learning. OpenMV is particularly well-suited to gesture recognition applications, as it allows for real-time image analysis on a microcontroller, without requiring the use of a separate computer [16]. OpenCV, on the other hand, is an open-source computer vision library that is widely used in academic research and commercial applications. It is designed to be used with a range of programming languages, including Python, C++, and Java, and provides a range of algorithms for image processing and machine learning. OpenCV is particularly useful for more complex gesture recognition applications, where a larger amount of data and more complex algorithms are required. Both OpenMV and OpenCV have been used in a range of gesture recognition projects, from simple hand gesture systems to complex applications. In many cases, these projects involve using machine learning algorithms to train a model to recognize different gestures based on input images or video streams [11-12].

While OpenMV and OpenCV offer several advantages for gesture recognition projects, they also have some limitations and consequences that need to be considered. One of the major consequences of using OpenMV and OpenCV for gesture recognition is the need for sufficient training data to train the machine learning model. This requires significant time and effort to capture and label the data, as well as to fine-tune the model for optimal performance [18]. Additionally, setting up and using these platforms efficiently requires a certain degree of technical know-how, which can be difficult for beginners or people without programming knowledge. Another potential disadvantage is the limited processing power and memory of microcontrollers like OpenMV, which can restrict the complexity and accuracy of the model. This means that some gestures may be misinterpreted or not recognized at all, which could lead to errors in the system.

V. PROPOSED SYSTEM

Gesture recognition using Wio Terminal and TinyML is a low power consumption project that involves connecting the device to a computer using USB and controlling it using CodeCraft [19-20]. The CodeCraft Seed Studio application is used to train the model with the help of the Edge Impulse platform. A dataset of different hand gestures is fed to the machine learning algorithm, which is optimised using TinyML. The Wio Terminal is equipped with a 3-axis accelerometer (LIS3DHTR) consisting of capacitors to detect hand gestures [13]. It also includes an infrared emitter, light sensor, Bluetooth, Wi-Fi, microphone, buzzer, and two multifunctioning groves. The model then analyzes the gesture made by the user and compares it with the trained datasets to recognize the pattern. This research demonstrates the viability of combining TinyML with Wio Terminal for real-time gesture detection applications. The low power consumption of TinyML and the Wio Terminal makes it ideal for edge computing applications, where the processing power of a device is crucial [14]. This allows the device to operate without the need for an internet connection, providing greater security and privacy for sensitive data. The project highlights the ease with which the Wio Terminal and TinyML can be used to build a machine learning application with minimal hardware requirements, making it accessible to a wider audience. With the increasing demand for edge computing and the Internet of Things, this project serves as a foundation for future research in the field of deploying machine learning directly on the microcontroller boards and thus making intelligent controllers.The figure 1.2 here showcases the parts of Wio Terminal.

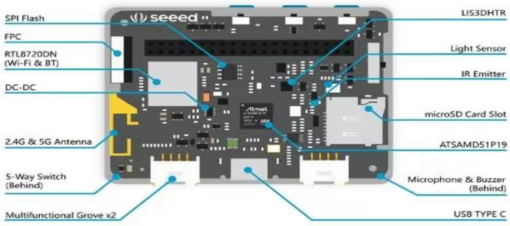

Figure 1.2: Schematic Diagram of Wio Terminal

VI. WORKING METHODOLOGY

1) Dataset collection:

A dataset of hand gesture movements is collected by capturing sensor data using the Wio Terminal's accelerometer and gyroscope. Figure 1.3 shows the goal setting of labelling gestures in the Wio terminal.

2) Dataset preprocessing:

In order to prepare the acquired dataset for machine learning model training, pertinent characteristics from the sensor data are extracted and converted.

3) Model training:

On the preprocessed dataset, a Convolutional Neural Network (CNN) model is trained using the CodeCraft EdgeStudio.

4) Model conversion:

Henceforth, the model that was trained using the suitable ML algorithm is now converted for deploying directly in Wio Terminal.

5) Real-time recognition of gestures:

The deployed model is used to classify the detected hand gestures in real-time using the Wio Terminal's sensors, and the corresponding action is displayed on the LCD screen.

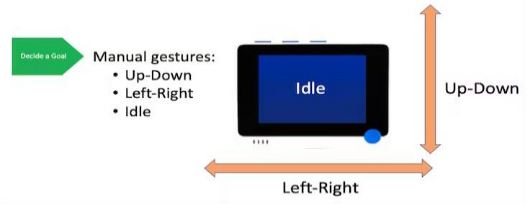

Figure 1.3: Goal setting

VII. RESULT

The result of Gesture recognition using TinyML and Wio Terminal is a functional system that enables real-time hand gesture recognition using sensors and microcontrollers. The system is capable of accurately recognizing and classifying hand gestures in real-time using the Wio Terminal's accelerometer and gyroscope sensors. The recognized gestures are then displayed on the LCD screen, allowing for easy feedback and interaction [15], [17]. The following figures 1.4, 1.5 and 1.6 respectively shows the recognized gestures displayed on Wio terminal's LCD screen.

The system's compact design and efficient use of resources make it a practical solution for various applications in fields such as robotics, human-computer interaction, and assistive technology. For example, the system could be integrated into a robotic arm to enable intuitive control using hand gestures, or into a wearable device to enable hands-free interaction with a computer or other devices.

Overall, the result of Gesture recognition using TinyML and Wio Terminal demonstrates the potential of using machine learning and microcontrollers to develop practical solutions for real-time gesture recognition using sensors, while making intelligent systems at very low cost.

Figure 1.4: Left Right gesture recognition (Output 1)

Figure 1.5: Idle gesture recognition (Output 2)

Figure 1.6: Up Down gesture recognition (Output 3)

VIII. CONCLUSION

The utilisation of TinyML on microcontrollers and IoT devices for machine learning has revolutionised the industry. It allows large amounts of data to be analysed in an optimised way, while using minimal power. The technology provides the ability to manage and process data locally, which reduces the susceptibility of sensitive data. Furthermore, TinyML is cost-effective, and operates without the need for internet connectivity. This makes it particularly useful for applications in the Web3 space, where spatial computing is a critical component. Moreover, in addition to its low-power advantages, TinyML can run complex machine learning algorithms without requiring internet connectivity. The gesture recognition project that employs TinyML and Wio Terminal is a testament to how the technology is building a bridge between humans and machines, and enabling real-time gesture recognition for applications in various fields such as human-computer interaction, robotics, gaming, and spatial computing.

REFERENCES

1. *TinyML: A Systematic Review and Synthesis of Existing Research. Hui Han, et al*
2. *Capacitive Sensing Based On-board Hand Gesture Recognition with TinyML. Sizhen Bian , et al*
3. *A Real-Time Hand Gesture Recognition Method. Yikai Fang, et al*
4. *Hand Gesture Recognition Based on Computer Vision: A Review of Techniques. Ali Al-Naji , et al*
5. *Towards energy-aware tinyML on battery-less IoT devices. Adnan Sabovic, et al*
6. *Sizhen Bian, et al 2019. Passive Capacitive based Approach for Full Body Gym Workout Recognition and Counting. In 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom. IEEE, Kyoto, Japan, 1–10.*
7. *Jun Rekimoto. 2001. Gesturewrist and gesturepad: Unobtrusive wearable interaction devices. In Proceedings*

Fifth International Symposium on Wearable Computers. IEEE, Zurich, Switzerland, 21–27.

8. *Hoang Truong, et al 2018. Capband: Battery-free successive capacitance sensing wristband for hand gesture recognition. In Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems. ACM, Shenzhen, China, 54–67.*
9. *Sizhen Bian, et al 2019. Wrist-Worn Capacitive Sensor for Activity and Physical Collaboration Recognition. In 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). IEEE, Kyoto, Japan, 261–266.*
10. *Memo, et al "Exploiting Silhouette Descriptors and Synthetic Data for Hand Gesture Recognition", STAG: Smart Tools & Apps for Graphics, 2015.*
11. *Erol, Ali, et al. "Vision-based hand pose estimation: A review." Computer Vision and Image Understanding 108.1-2 (2007): 52-73.*
12. *Rohith, et al "Hand gesture recognition in real time using IR sensor." International Journal of Pure and Applied Mathematics 114.7 (2017): 111-121.*
13. *Ha, et al "Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors." 2016 International Joint Conference on Neural Networks (IJCNN). IEEE, 2016.*
14. *Eggimann, et al., 2019, October. Low Power Embedded Gesture Recognition Using Novel Short-Range Radar Sensors. In 2019 IEEE SENSORS (pp. 1-4). IEEE.*
15. *C. R. Banbury et al., "Benchmarking TinyML systems: Challenges and direction", Proc. 1st Int. Workshop Benchmarking Machine Learning Workloads Emerging Hardware - 3rd Conf. Machine Learning and Systems (MLSys), 2020*
16. *Khan, et al. Use hand gesture to write in air and recognize with computer vision. IJCSNS 2017, 17, 51.*
17. *B. Sudharsan et al., "TinyML benchmark: executing fully connected neural networks on commodity microcontrollers", IEEE World Forum on Internet of Things, pp. 19-21*
18. *R. David et al., "TensorFlow Lite Micro: embedded machine learning on tinyml systems", 2020.*
19. *Y. Maret, et al "Real-time embedded system for gesture recognition", 2018 IEEE International Conference on Systems Man and Cybernetics (SMC).*
20. *Alajlan, et al. TinyML: Enabling of Inference Deep Learning Models on Ultra-Low-Power IoT Edge Devices for AI Applications. Micromachines 2022, 13, 851.*