UCL Centre for Advanced Spatial Analysis

Connected Environments

Deep Learning for Sensor Networks

CASA0018

Final Report

Sign Language interpretation gloves

University College London

Dankao Chen:	24067882
Date:	20/4/2025
Words:	1615
GitHub:	https://github.com/Reikimen/CASA0018-Gloves-Edge-AI

Contents

1 Introduction

1.1 Project Overview

The Sign Language Recognition Glove project aims to develop a wearable device that translates sign language by recognising hand positions in real time. According to the World Health Organisation, about 430 million people have varying degrees of hearing impairment, including 34 million children (World Health Organization, 2023). Sign language, as a visual language, has different variants and dialects across the globe. While significant advances have been made in visual recognition technology, wearable device-based solutions offer unique advantages, especially in terms of privacy protection, environmental adaptability and real-time usage. In this project, a flexible sensor is combined with an Inertial Measurement Unit (IMU) to capture finger flexion and hand movement to recognise specific sign language postures and provide real-time sign language interpretation.

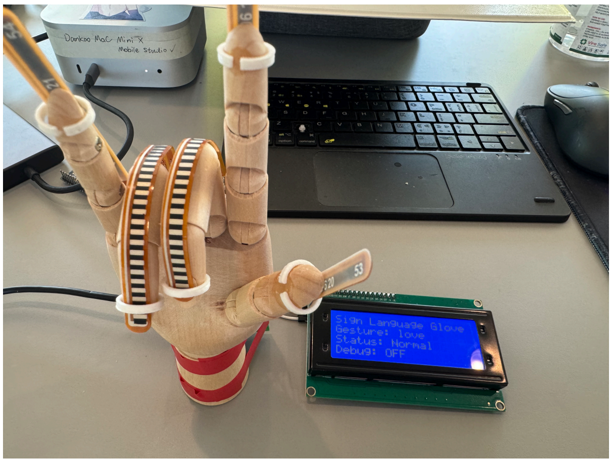

Figure 1: Project Overview

1.2 Background research and related work

The history of research in sign language recognition technology dates back several decades, with the early days relying primarily on visual recognition systems (Min et al., 2021). These systems captured hand movements via cameras and applied computer vision algorithms for recognition. Despite the success of visual approaches in experimental settings, they still face many challenges in real-world applications, such as changing lighting conditions, background complexity, and occlusion problems. In contrast, sensor-based glove solutions offer a different technological path. Sensor gloves avoid the environmental limitations of visual recognition by directly measuring finger bending angle and hand posture. This approach protects privacy, works in any lighting conditions, and provides more accurate data collection.

In recent years, with the development of edge AI technologies, it has become possible to deploy machine learning models to resource-constrained embedded devices. Advances in TinyML (Tiny Machine Learning) have allowed complex models to run on microcontrollers, opening up new possibilities for wearable assistive technologies (Diab and Rodriguez-Villegas, 2022). The emergence of open source platforms such as Edge Impulse has further lowered the development barriers, making it easier for developers to train and deploy machine learning models to embedded devices (Hymel et al., 2022).

2 System integration and implementation

2.1 User Interface

The user interface design consists of two parts: the LCD display and the serial command interface.The LCD display provides intuitive visual feedback on the currently recognised gestures, sensor states and system messages. The serial command interface, on the other hand, provides a rich set of debugging and control functions, including viewing sensor data, displaying a list of supported gestures, and visualising the bending angle of the finger, which facilitates system tuning during development and testing.

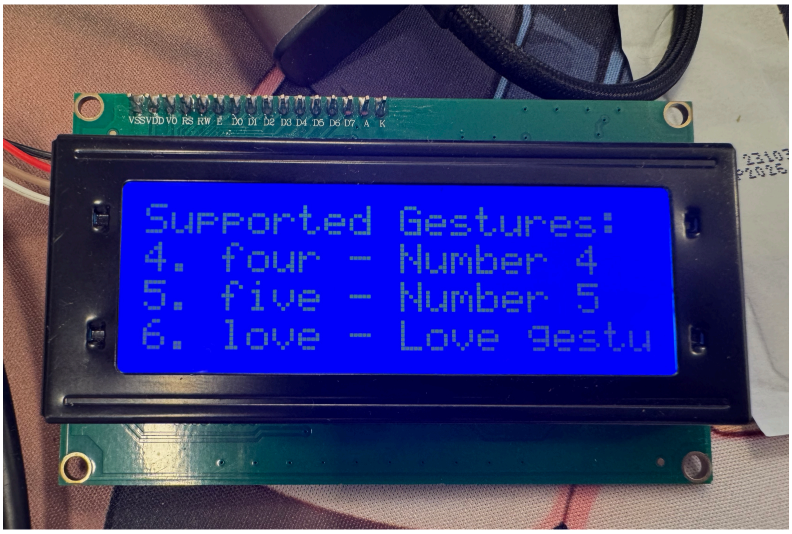

Figure 2: UI on LCD

2.2 Hardware Design and Implementation

The sensor selection for this project combines five flex sensors and an inertial measurement unit (IMU). The five flexible sensors are mounted at the thumb, index, middle, ring and little finger positions of the glove and are used to measure finger flexion. These sensors are able to accurately capture the flexion and extension of the fingers in various sign language postures, which is key to recognising basic sign languages. Meanwhile, the LSM9DS1 inertial measurement unit that comes with the Arduino Nano 33 BLE provides acceleration and gyroscope data to detect hand orientation and movement, providing additional information for complex sign language recognition that requires dynamic gestures.

The choice of hardware platform, Arduino Nano 33 BLE, is also based on a number of considerations. Firstly, its nRF52840 processor provides enough computing power to process real-time sensor data and perform machine learning inference; secondly, the built-in Bluetooth functionality offers the possibility of extended wireless communication in the future; and thirdly, the on-board IMU simplifies hardware integration. These features make the Arduino Nano 33 BLE an ideal controller for sign language recognition gloves.

2.3 Data acquisition and processing

The data acquisition process begins with the Arduino's analogue-to-digital converter (ADC) reading the change in resistance of the flex sensor. The system samples at a frequency of 50Hz, which is sufficient to capture changes in hand movements without overloading processing resources. The program reads the sensor values of the five fingers sequentially through predefined pin configurations (FLEX_PIN_THUMB to FLEX_PIN_PINKY) and collects the IMU data at the same time to ensure the synchronisation of the multi-sensor data.

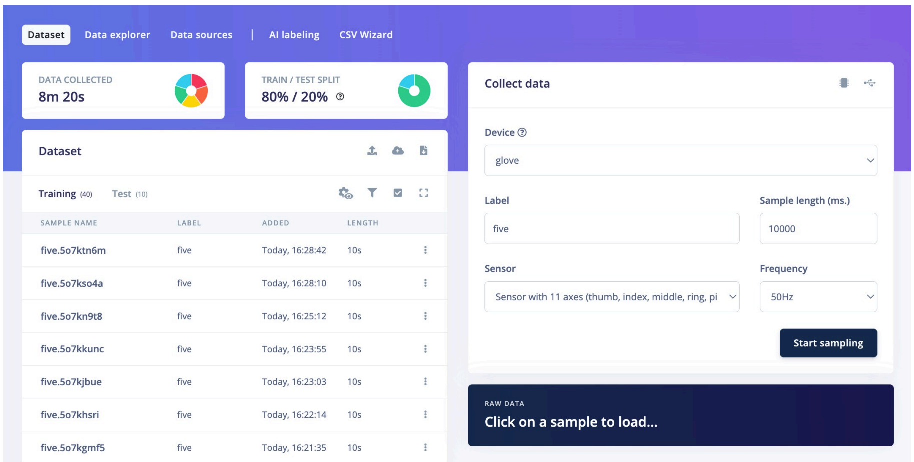

Figure 3: The 8:2 Collected Dataset

The data processing uses a sliding window technique to manage a sample window of size 50 through the "updateDataWindow" function. This approach allows the system to compute a series of statistical features that reflect the temporal characteristics of the gesture. Together, these features form the input feature vector of the machine learning model.

In order to improve recognition stability, the data is preprocessed in the system. A low-pass filter (coefficient = 0.3) is used to smooth the sensor readings and reduce the effect of noise. In addition, the system handles data normalisation to ensure that the eigenvalues are within the appropriate range for easy processing by machine learning models. For example, the code implements a mapping function "calculateBendPercentage" from the raw ADC value to the bend percentage. This function handles the problem of nonlinear response and individual differences in the sensor by linearly interpolating the pre-measured straight extension (FLEX_STRAIGHT_ADC) and full bend (FLEX_BENT_ADC) values.

2.4 Model architecture and training

This project utilises the Edge Impulse platform for model development, which is designed for embedded machine learning and provides a complete workflow from data collection to model deployment.

Feature selection is a key decision in model design. The system computes 7 statistical features for each flexible sensor, generating a total of 35 features (5 sensors x 7 statistics). For each sensor, the system computes seven statistical features: mean, minimum, maximum, root mean square, standard deviation, skewness and kurtosis. This comprehensive feature set captures both static postures and dynamic patterns of change in finger flexion. These statistical features provide a higher level of information representation compared to raw time series data.

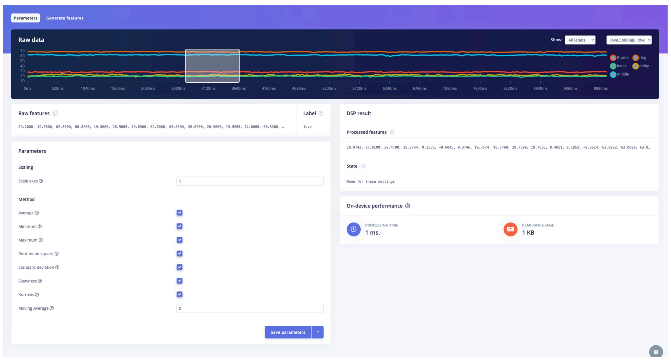

Figure 4: 35 features Flatten

The training process includes steps such as data collection, feature extraction, model training and validation. The Neural Network settings (Input layer, Dense layers, and Output layer) and training performance is as follows:

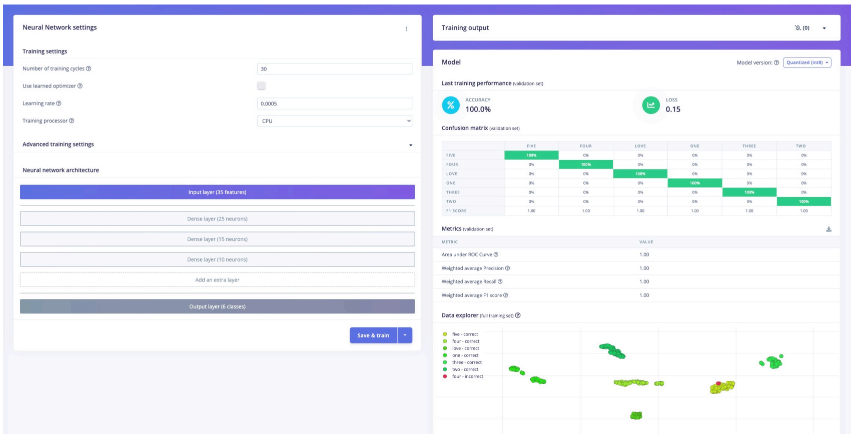

Figure 5: Neural Network settings and training performance

Model optimisation takes into account a variety of factors including accuracy, latency and memory footprint. The models are quantified and optimised to fit the constraints of the embedded environment. The system (Arduino part) implements an efficient inference calling mechanism that performs inference through the run_classifier function and provides feature data through the callback function get_signal_data.

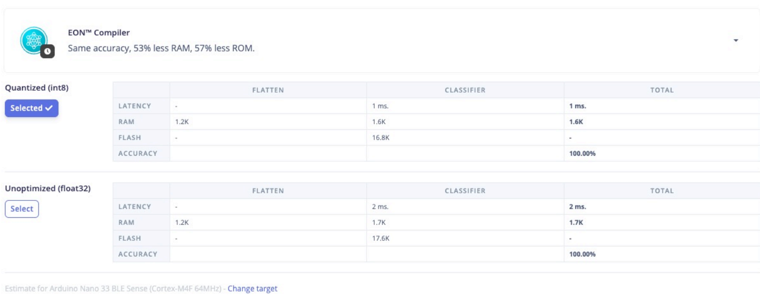

Figure 6: Model optimisation (Quantize)

For the recognition part, the project imported the library named Sign-Language-Glove_inferencing.h generated from Edge Impulse, which contains the model definition and inference function. 60% threshold was set for the Arduino part of the system.

2.5 Experimental Results and performance analysis

In order to improve the recognition stability, the system implements a continuous recognition mechanism. The system outputs the result only when the same gesture is recognised several times in succession and the confidence level exceeds a threshold. This design reduces transient misrecognition and improves the reliability of the system. This project also did a lot on optimization (different versions):

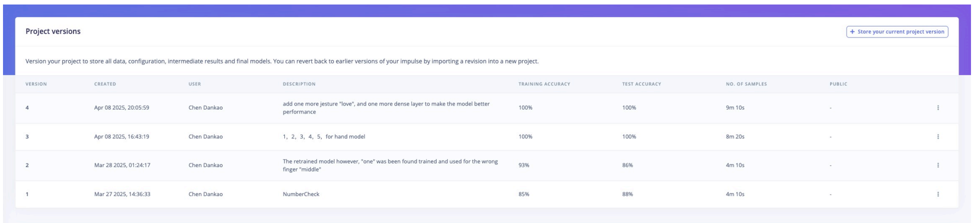

Figure 7: Performance Optimization (diff versions)

Here is the final result for the Model testing, which indicates the high performance:

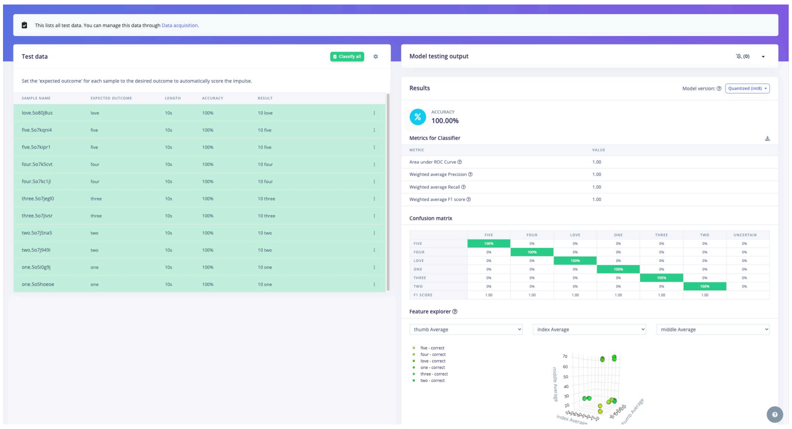

Figure 8: Model testing result (final ver)

Here is one of the live classification result (example for love):

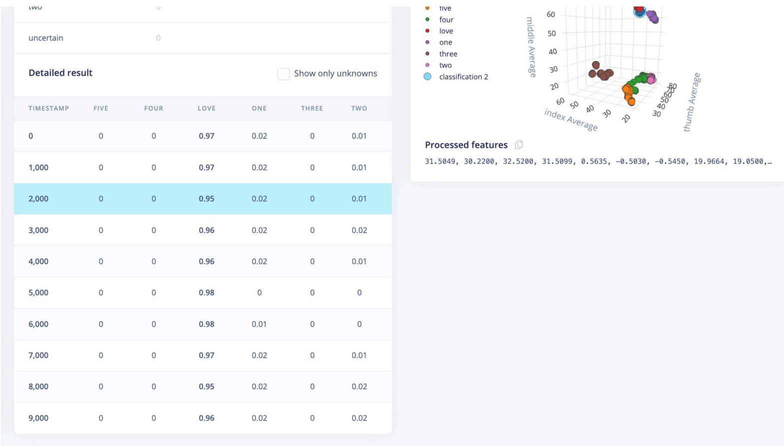

Figure 9: live classification (love)

The system performance is quite well. In addition, The system also implements ”no gesture” detection, which clears the currently recognised gesture when the confidence level continues to fall below the threshold for a period of time, preventing the error state from persisting.

3 Key challenges and solutions

Matching the model input format was a major challenge in the implementation.The model generated by Edge Impulse expects a specific format of input data, and needs to accurately convert the sensor data into the feature vectors required by the model. The system solves this problem with the ”prepareFeatures” function, which computes statistical features for each sensor and populates the feature array in the order expected by the model.

Sensor drift and calibration is another notable challenge. Flexible sensors may change output characteristics during use due to material fatigue, temperature changes, and other factors. The system addresses this issue with configurable calibration values (FLEX_STRAIGHT_ADC and FLEX_BENT_ADC) and adaptive calculation functions. The user can update these values through the calibration procedure, adapting to changes in the sensor over time. In addition, the implementation of a low-pass filter helps to reduce short-term fluctuations and improve reading stability.

Optimisation in resource-constrained environments is an ongoing challenge during the design process. The Arduino Nano 33 BLE, while powerful, still has limitations in RAM and computational power, especially when performing machine learning tasks. The system addresses this challenge through a variety of strategies: Efficient sliding Windows, task time segmentation, algorithm optimization and conditional compilation to save resources.

The balance between real-time and accuracy is a central challenge in system design. Sign language recognition requires fast response, but improving accuracy usually requires more complex models and more computation. The system strikes the balance in a number of ways: the sampling rate (50Hz) and inference interval (300ms) are carefully chosen to ensure sufficient data acquisition frequency while avoiding over-inference; ”stableCount” is implemented to reduce recognition jitter and improve user experience; and confidence thresholds are set to filter out low-confidence recognition results. Together, these mechanisms ensure that the system provides reliable recognition results while maintaining real-time response.

4 Critical reflection

4.1 Project Limitation

Although the project has achieved some success, there are still several limitations. Firstly, the number of gestures currently supported is limited (only six) and only static gesture recognition is currently supported, making it difficult to cover the complete sign language vocabulary. Expanding the vocabulary will require more training data and potentially more complex model architectures. Secondly, hand shapes, gesture expressions and usage habits vary greatly among different users, and the system needs stronger adaptive mechanisms. Next, environmental sensitivity is also an issue, as the curvature sensors currently used will age over time and affect recognition stability and reliability. Last but not least, the project suffers from certain performance bottlenecks. The performance bottleneck comes from 2 main sources: computational constraints, and sensor resolution. The processing power of the Arduino platform limits the model complexity and inference frequency that can be used. The poor resolution and accuracy of the flexible sensor constrains the system's ability to distinguish between small gesture differences.

4.2 Commercial feasibility

In terms of commercial feasibility, the project demonstrates potential as an assistive technology product. Cost estimates need to take into account factors such as hardware components (Arduino, sensors, LCD display), assembly and production costs. The market potential is vast and includes not only the hearing impaired, but also sign language learners, educational institutions and special scenarios where silent communication is required. The possible impact of the project is not only in terms of economic value, but also in terms of improving the quality of life and social participation of the hearing impaired.

4.3 Future Works

Future direction of improvement includes a number of aspects: firstly, expanding the gesture vocabulary to include more daily-used sign language words and phrases; secondly, implementing an adaptive calibration mechanism that automatically adjusts the sensor parameters to adapt to different users and environmental conditions; and thirdly, optimising the machine learning model and exploring more efficient model architectures or compression techniques. As for the long-term goal of the project, it may include integrating more sensor types, such as haptic feedback or myoelectric sensors, to enhance the system capabilities while promoting user comfort.

5 Conclusion

The Sign Language Interpreting Glove project successfully combined sensor technology, embedded systems, and machine learning to create a real-time sign language recognition prototype. The project provided initial validation of the feasibility of embedded sign language recognition, demonstrating that relatively complex gesture recognition tasks can be achieved even on resource-constrained platforms. Although there is still space for improvement in the current version, it lays a solid foundation for more advanced sign language recognition systems in the future, and is an important step towards building a more inclusive and accessible communication environment.

References

Diab, M. S. and Rodriguez-Villegas, E. (2022). Embedded machine learning using microcontrollers in wearable and ambulatory systems for health and care applications: A review. *IEEE Access*. 10, pp. 98450–98474.

Hymel, S., Banbury, C., Situnayake, D., Elium, A., Ward, C., Kelcey, M., Baaijens, M., Majchrzycki, M., Plunkett, J., Tischler, D., et al. (2022). Edge impulse: An mlops platform for tiny machine learning. *arXiv preprint arXiv:2212.03332*.

Min, Y., Hao, A., Chai, X., and Chen, X. (2021). "Visual alignment constraint for continuous sign language recognition". In: *proceedings of the IEEE/CVF international conference on computer vision*, pp. 11542–11551.

World Health Organization (2023). *Deafness and hearing loss*. Accessed: 17 April 2025. Available from: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss.