SUGISAKA MASANORI

Journal of Robotics, Networking and Artificial Life
Vol. 10(4); March (2024), pp. 313–317
ISSN (Online): 2352-6386; ISSN (Print): 2405-9021
https://alife-robotics.org/jrnal.html

Research Article

MCU Based Edge Computing Platform for Liquid Level Measurement

Chung-Wen Hung[1], Chun-Liang Liu[1], Tai-Hsuan Wu[1], Chau-Chung Song[2]
[1]*National Yunlin University of Science and Technology, Taiwan, 123 University Road, Section 3, Douliou, Yunlin 64002, Taiwan, R.O.C*
[2]*National Formosa University, Taiwan, 64 Wenhua Road, Huwei Town, Yunlin County 632301, Taiwan, R.O.C*

ARTICLE INFO

Article History

Received 15 November 2023
Accepted 31 July 2024

Keywords

Label smoothing
Edge computing
Audio process
Hyperparameter

ABSTRACT

An MCU(microcontroller unit)-based edge computing system for assessing the height of liquids in containers in this paper. This system uses a solenoid valve to strike the bottle and a microphone to capture sound waves. Signals are transformed from the time domain to the frequency domain by FFT (Fast Fourier Transform) and used to predict the water level by using AI (artificial intelligence) model. A dynamic label smoothing method enhances label correlation, and an ANN (Artificial Neural Network) model is employed on the MCU for classification. The system accurately predicts water levels from 200 to 250 milliliters at 1-milliliter intervals. Hyperparameter optimization balances accuracy with MCU memory and computational constraints. Experimental results show that the system achieves an accuracy of 81% under the limits of edge computing, verifying its effectiveness in liquid level measurement applications.

1. Introduction

The rapid expansion of artificial intelligence has led to a multitude of applications across consumer and industrial sectors [1]. Edge computing [2], [3], [4] has become a critical focus area, with particular emphasis on utilizing endpoints for model computation to minimize network latency and expenses. In scenarios where traditional sensors are impractical or costly, exploring alternative AI-based measurement techniques becomes essential.

One such application involves analysing the sound produced by impacted containers, such as bottles, where liquid levels influence vibration frequencies. As noted in [5], sound-based water level determination is dependent on specific vessel geometries, making AI models a viable solution for this measurement approach.

Considering the resource constraints of endpoint platforms, it is crucial to select appropriate AI models and optimize hyperparameters for edge computing. ANN models are known for their simplicity and effectiveness in classification tasks, making them a promising candidate model for this paper [6]. Furthermore, Bayesian optimization stands out as an efficient method for hyperparameter tuning.

The paper introduces an edge computing system employing an MCU to measure liquid levels, employing the Artery AT32F415 [7] consumer MCU for all computations and controls, including data acquisition. By utilizing one-dimensional audio data, the proposed AI model maps audio features to different groups through model learning, effectively reducing memory footprint. The system incorporates a data sampling function during the training phase, transmitting collected information to a PC for hyperparameter optimization and model training.

Methodology

1.1. *SYSTEM STRUCTURE*

Fig. 1 presents the system diagram, which incorporates two micro switches to initiate the process of knocking, sampling, and edge computing. These switches are crucial for maintaining consistent hitting conditions, including distance and position.

The MCU serves as the core of the system, orchestrating the entire operation and providing edge

Corresponding author E-mail: wenhung@yuntech.edu.tw, clliu@yuntech.edu.tw, m11112080@ yuntech.edu.tw, ccsong@nfu.edu.tw

computing functionalities. For training purposes, the system utilizes UART (Universal Asynchronous Receiver/Transmitter) protocol to transfer data between the MCU and PC. This communication channel allows for the transmission of sound samples from the microcontroller to the computer for model training.

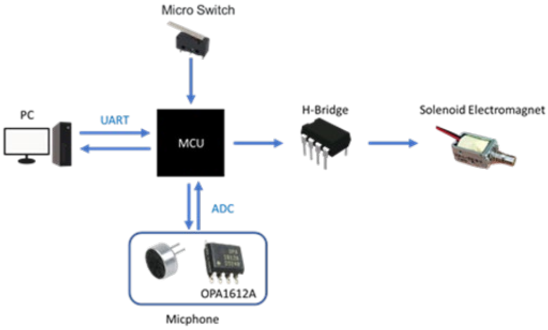

Fig. 1 The system diagram

An H-Bridge [8] is used to drive the knocking mechanism (a solenoid electromagnet), enabling precise control over its extension and retraction movements so it can strike the water bottle. A microphone is used to capture the knocking sound. The capacitive microphone converts sound waves into voltage, which is then amplified in amplitude. The voltage offset is calibrated using an operational amplifier (OPA) circuit [9] to ensure compatibility with the input range of the ADC (Analog to Digital Converter).

1.2. *DATA ANALYSIS AND REPROPCESSING*

Before modeling, data analysis is paramount. The converted sampling data is transmitted to a PC via UART for analysis and training. Audio voltage sampling occurs at 10 kHz, with 2048 points collected per trigger. Fig. 2 displays time domain waveforms for various water levels, showing no saturation or truncation, indicating appropriate sampling rate and length for this application.

To facilitate algorithm development, Fourier transform is applied to convert time domain data into the frequency domain, as illustrated in Fig. 3. Notably, the first spike's frequency decreases as water volume increases, aligning with auditory observations. The analysis utilizes frequency information between 83 Hz and 1464.8 Hz for water level identification.

This study utilizes FFT for data pre-processing. To mitigate the impact of abnormal waveforms and optimize system performance, each dataset undergoes min-max normalization. This normalization technique scales the sound amplitude values to a range of 0 to 1 using the provided Eq. (1). In the formula, X represents the original data, X_{nom} is the normalized result, and X_{max} and X_{min} are the maximum and minimum values in the dataset respectively.

$$X_{nom} = \frac{X - X_{min}}{X_{max} - X_{min}} \in [0,1] \quad (1)$$

For the classification task, an ANN model was selected. The ANN, with its simplicity, classification efficiency, and lower memory and computational resource requirements, proves more suitable for deployment on edge computing platforms such as MCU compared to other AI models. This approach provides an efficient and resource-friendly solution for water level classification.

This paper employs an electromagnetic striking device to generate sound samples from glass bottles containing varying amounts of water. The water volume ranges from 200 ml to 250 ml, with samples taken at 1 ml intervals. The collected sound data undergoes pre-processing before being used to train an artificial neural network model. Notably, the model training utilizes only the sampling data from 200, 210, 220, 230, 240, and 250 ml volumes.

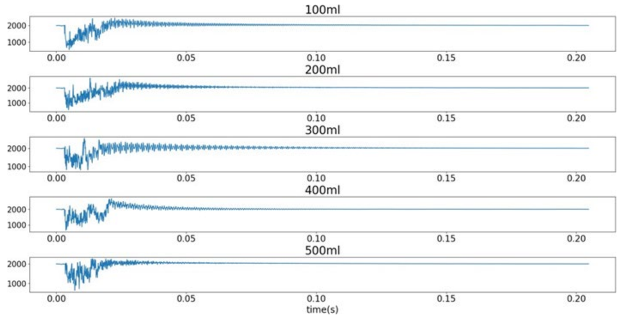

Fig. 2 Audio time domain at different water levels

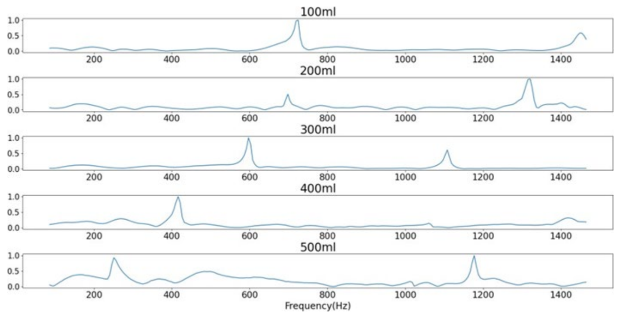

Fig. 3 Frequency domain at different water levels

1.3. *LABEL SMOOTHING*

In the training of neural network models, one-hot encoding is usually used to mark different samples for classification purposes. One-hot encoding sets only the correct sample classification to 1, while other classifications are set to 0, as Eq. (2). i is represented the target label to be annotated, and $target$ denotes the true category of the target label. The target label is set to 1, and all others are set to 0.

$$Label[i] = \begin{cases} 1, & when\ i = target \\ 0 & \end{cases} \quad (2)$$

One-hot encoding causes the model to focus on increasing the prediction probability of the true class during training. However, it cannot represent any relationship between categories and ignores their similarities. In addition, for datasets with small amounts of data or large models, overfitting may occur as the number of training iterations increases.

In order to solve this problem, this paper uses label smoothing for annotation. Unlike traditional label smoothing shown in Eq. (3), which adjusts the original definite true label from 1 to a value slightly less than 1 and distributes the reduced portion(K) to other classes($N-1$), this paper proposes a dynamic label smoothing [10], [11], [12] method.

$$Label[i] = \begin{cases} 1-K, & when\ i = target \\ \dfrac{K}{(N-1)} & \end{cases} \tag{3}$$

This method does not distribute the same label value equally among all incorrect labels. Instead, it dynamically assigns tag values based on the correlation between each tag and the correct tag. The correct label is assigned the highest label value, while other categories receive their respective label values based on how related they are to the correct label, as shown in Eq. (4). Here, K and S are coefficients that need to be preset. The value of K indicates how much the correct label will receive, while S affects the value at which the label values for the incorrect labels decrease. The variable n represents the distance from the correct label. In Fig. 4, the label assignment scenarios under different K and S values can be observed.

$$Label[i] = \begin{cases} 1-K, & if(i = target) \\ \dfrac{K}{(S\times n!)}, & else \end{cases} \tag{4}$$

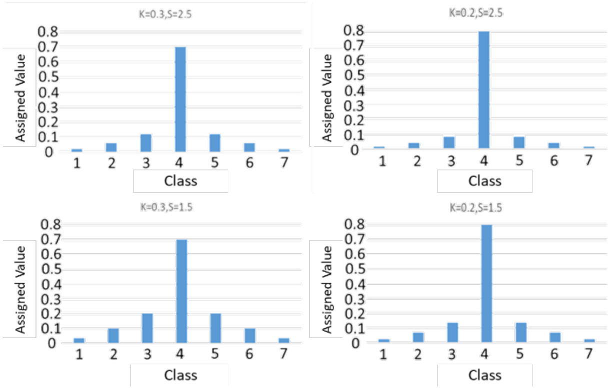

Fig. 4 Label value distribution under different ***K*** and ***S*** coefficients

$$Label_{Nor}[i] = \frac{Label[i]}{\sum_{i=1}^{N} Label[i]} \tag{5}$$

To ensure that the labels sum to 1, the overall labels are normalized using Eq. (5). N represents the number of label categories, and $Label_{Nor}$ is the modified label value. This normalization preserves the original proportional relationship between values while making the sum of all label values equal to 1.

This approach prevents the model from becoming overly confident in predicting a single class, thereby improving its generalization ability. It also enables the model to consider the potential correlations between different classes, dynamically adjusting the label values based on these correlations.

1.4. *HYPERPARAMETER OPTIMIZATION*

In deep learning algorithms, the choice of hyperparameters determines the size and performance of the artificial intelligence model. These parameters, such as the number of in each layer, hidden layers, and activation functions directly affect the architecture of the model and its ability to generalize to new data. Traditionally, researchers prioritize the hyperparameter combination that yields the highest accuracy for implementation in MCU. However, in edge computing systems, resource limitations become a crucial factor. Therefore, it's essential to strike a balance between meeting accuracy requirements and selecting parameters that align with the available computational resources. This approach ensures optimal performance within the constraints of edge devices.

Bayesian optimization [13] is an effective method for solving black-box optimization problems. Unlike traditional approaches such as random search or grid search, Bayesian optimization constructs a Gaussian Process model of the objective function as a prior distribution. This Gaussian Process efficiently utilizes past evaluation results and predicts uncertainties, guiding subsequent search processes. By leveraging the probabilistic model, Bayesian optimization can effectively explore the parameter space and exploit promising regions, leading to more efficient optimization compared to conventional methods.

2. Results and Discussion

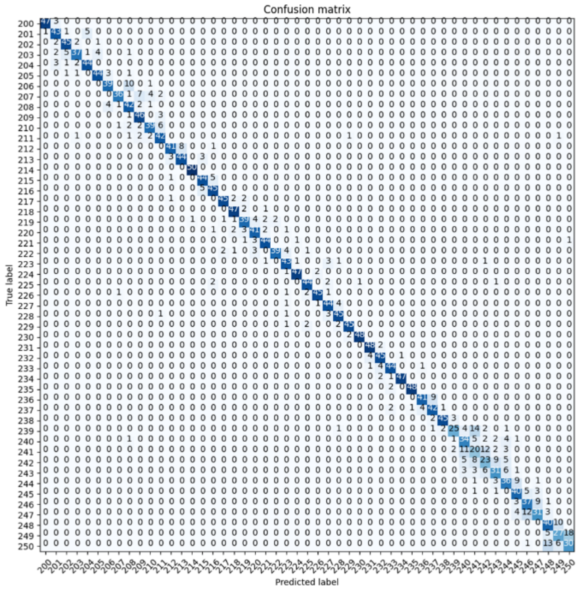

Fig. 5 Confusion matrix smoothed by dynamic labels

The water level prediction model was trained using dynamic label smoothing, with 200, 210, 220, 230, 240, and 250 ml as the primary labels. After training, the confidence scores of models for these six labels were used to extend predictions to 1 ml intervals, achieving finer water level predictions. Fig. 5, 4, 3 shows the confusion matrix for this model. While some samples in the red frame area deviate from their true label, most predictions are clustered within ±5 ml of the actual label. After hyperparameter optimization, the model accuracy was 82%, indicating that the model performed well. The entire model requires only 76KB of MCU memory.

Mean Absolute Error (MAE) calculates the average absolute difference between predicted and actual values, as shown in Eq. (6)

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | y_i - \hat{y}_i | \tag{6}$$

Where n is the number of samples, y_i is the actual value of the i -th sample, and $\hat{y}_i$ is the predicted value of the i -th sample. MAE is easily understood and interpreted. It can directly reflect the average deviation between predicted and actual values. A smaller MAE indicates more accurate predictions. However, it weighs all errors equally, failing to distinguish between large and small errors, which can lead to incomplete judgments in scenarios with extreme errors.

To complement MAE, this experiment also calculated the Root Mean Square Error (RMSE), which is the square root of the average of squared differences between predicted and actual values. RMSE helps assess the impact of extreme errors. Its formula follows Eq. (7), where the variables are defined as in the MAE equation.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{7}$$

Compared to MAE, RMSE is more sensitive to extreme errors as squaring errors amplifies the impact of large deviations, effectively penalizing large errors more heavily.

These results indicate good overall predictive performance of the model, but the presence of some larger errors leads to a relatively high RMSE. This aligns with the observations from the confusion matrix.

3. Conclusion

This paper presents an MCU-based edge computing platform for liquid level measurement. The system encompasses not only physical components such as the MCU, audio signal pre-processing circuit and striking mechanism but also provides a detailed explanation of the development process. This includes the normalization of sampled data, the introduction of label smoothing techniques, and the optimization of hyperparameters.

In the experimental setup, glass bottles were utilized, which resulted in more pronounced energy variations in the impact sounds. This choice of container material enhanced the system's ability to differentiate between various liquid levels based on acoustic signatures.

After multiple tests, it was determined that using 200, 210, 220, 230, 240, and 250 ml as model inputs could effectively predict results across the 200-250 ml range at 1 ml intervals. The experimental results yielded an accuracy of 81%, while requiring only 76KB of MCU memory. These findings demonstrate the system's viability and confirm its satisfactory measurement precision, highlighting the efficient use of limited computational resources in edge computing applications.

References

1. C. W. Hung, S.X. Zeng, C.H. Lee, W.T. Li. End-to-End Deep Learning by MCU Implementation: An Intelligent Gripper for Shape Identification. Sensors, 2021, Volume 21(3):891.
2. C. Hegde, et al. Autotriage-an open source edge computing raspberry pi-based clinical screening system. medRxiv [Internet], 2020, p. 2020.04.09.20059840. Available from: medRxiv.
3. S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, A.Y. Zomaya. Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence. IEEE Internet of Things Journal, 2020, Volume 7(8):7457-7469.
4. C. W. Hung, K. Hiroyuki, J.R. Wu, C.C. Song. Low-Cost Indoor Localization Using Sound Spectrum of Light Fingerprints. In: Proceedings of the 2021 International Conference on Artificial Life and Robotics (ICAROB), 2021, pp. 89-94.
5. S. Birlasekaran, G. Ledwich. Use of FFT and ANN techniques in monitoring of transformer fault gases. In:

Proceedings of 1998 International Symposium on Electrical Insulating Materials. 1998 Asian International Conference on Dielectrics and Electrical Insulation. 30th Symposium on Electrical Insulating Materials. IEEE, 1998, pp. 75-78.
6. F. Foukalas, A. Tziouvaras. Edge artificial intelligence for industrial internet of things applications: an industrial edge intelligence solution. IEEE Industrial Electronics Magazine, 2021, Volume 15(2):28-36.
7. ARTERY Technology. AT32F415S DATASHEET [Internet]. June 2022 [V2.01]. Available from: ARTERY Technology.
8. Texas Instruments. OPA1612A DATASHEET [Internet]. March 2015 [Rev.c]. Available from: Texas Instruments.
9. ASIC. L9110H DATASHEET [Internet]. Available from: ASIC.
10. R. Müller, S. Kornblith, G.E. Hinton. When does label smoothing help? Advances in Neural Information Processing Systems, 2019, Volume 32.
11. C. B. Zhang, P.T. Jiang, Q. Hou, Y. Wei, Q. Han, Z. Li, M.M. Cheng. Delving deep into label smoothing. IEEE Transactions on Image Processing, 2021, Volume 30:5984-5996.
12. T. Lukov, N. Zhao, G.H. Lee, S.N. Lim. Teaching with soft label smoothing for mitigating noisy labels in facial expressions. In: European Conference on Computer Vision. Springer, 2022, pp. 648-663.
13. H. Alibrahim, S.A. Ludwig. Hyperparameter optimization: Comparing genetic algorithm against grid search and bayesian optimization. In: 2021 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2021, pp. 1551-1559.

Authors Introduction

Dr. Chung-Wen Hung

He was awarded a Ph.D. in Electrical Engineering from National Taiwan University in 2006. Currently, he holds a professorship at National Yunlin University of Science and Technology. His research focuses on AI, IIoT, and IoT applications.

Dr. Chun-Liang Liu

He earned his Ph.D. in Electrical Engineering from National Taiwan University of Science and Technology. He has since joined the faculty at National Yunlin University of Science and Technology, where he currently serves as an Assistant Professor

Mr. Tai-Hsuan Wu

He has obtained his bachelor's degree and is currently pursuing a master's degree in electrical engineering at National Yunlin University of Science and Technology.

Dr. Chau-Chung Song

He received the Ph.D. degrees in Electrical Engineering from National chiao tung university. Currently he is a Distinguished professor and Chair in National Formosa University