TINYML-BASED VIETNAMESE KEYWORD RECOGNITION FOR SMART HOME VOICE CONTROL

Duan Luong Cong*, **Cuong Chu Van***, **Minh Nguyen Ngoc***,
* Faculty of Electronics Engineering 1, Posts and Telecommunications Institute of Technology

***Abstract*—This study develops a TinyML-based voice control system specifically for the Vietnamese language, addressing the demand for localized smart home solutions. Unlike existing research focused on English, our work provides an accessible option for Vietnamese speakers. We collected a dataset comprising one wake-up keyword and eight common commands, integrating DSP algorithms and machine learning models on the ESP32-S3 MCU. The model, with 129,098 parameters, achieved an average accuracy of 94% and a real-time execution time of 368 ms for data windows of 500-1000 ms. This highlights the potential of TinyML in voice-controlled devices, offering a low-cost, Internet-independent solution that enhances data security and user privacy.**

***Index Terms*—Smart Home, Keyword Spotting, Voice Control, Endpoint Detection, TinyML, MFCCs.**

I. INTRODUCTION

Smart homes leverage advanced technologies, particularly Internet of Things (IoT) devices, to facilitate remote management of household systems. Adoption of these devices is rapidly increasing, with projections estimating over 500 million connected devices in the average home by 2025. This shift enhances energy efficiency, security, and convenience, enabling residents to automate routine tasks and create a more enjoyable living experience [1], [2].

In this context, voice control has become a valuable interface for smart home systems, offering convenience, speed, and natural interaction. By allowing hands-free operation, voice commands enable residents to manage devices effortlessly, significantly enhancing user experience. The rapid advancement of speech recognition technology further supports this trend, making voice control more accurate and responsive. As these technologies evolve, they provide a seamless means of interaction, addressing the growing demand for user-friendly automation solutions [3], [4], [5], [6].

Many applications utilize virtual assistants like Alexa and Google Assistant for voice control in smart homes, allowing users to manage devices through simple commands [7]. However, these systems face challenges, such as requiring stable Internet connections, exhibiting latency, and raising privacy concerns [8]. Despite these issues, research and development in voice control are expanding, particularly in creating multilingual systems for diverse user bases. This focus on inclusivity enhances accessibility for non-English speakers, addressing a critical gap in current solutions [9], [10], [11]. Local voice control offers significant advantages, including improved data security, independence from Internet connectivity, and rapid response times. These benefits make it an appealing alternative for users prioritizing privacy and reliability, with ongoing research exploring various applications that operate entirely on-device [12].

Tiny machine learning (TinyML) [13], which refers to deploying machine learning models on resource-constrained devices, plays a crucial role by enabling intelligent processing at the network's edge [14]. This technology is especially relevant for IoT devices, where efficient data processing is critical. TinyML is ideal for local voice control due to its high performance and energy efficiency, facilitating real-time voice recognition without continuous Internet connectivity. Its compatibility with embedded systems allows implementation across various smart home devices, enhancing overall user experience.

Recent studies have investigated TinyML applications in speech recognition, demonstrating its potential for efficient voice control in constrained environments [15], [16], [17], [18]. However, a significant gap exists regarding its implementation for the Vietnamese language, as most existing systems focus on English and other widely spoken languages. This limitation restricts accessibility for Vietnamese speakers, underscoring the need for further research [19], [20].

This study aims to address this gap by developing a localized voice control system using TinyML techniques tailored for the Vietnamese language. The methodology involves collecting a diverse datasets of Vietnamese keywords, training machine learning models, and deploying these models on micro-controller (MCU). The research's significance lies in its potential to enhance the usability and accessibility of smart home technologies for Vietnamese users, thereby contributing to the broader adoption of local voice control

Contact author: Minh Nguyen Ngoc
Email: minhnn@ptit.edu.vn
Manuscript received: 3/2025, revised: 4/2025, accepted: 5/2025.

systems in the region.

The exist of paper is organized as follows: Section 2 presents the methodology in detail, outlining data collection, pre-processing, model training processes, and implementation steps. Section 3 provides results and a comprehensive discussion of the findings.

II. METHODOLOGY

A. Experimental Context

To establish the experimental model for this study, we developed a voice control device based on keyword spotting for a room equipped with essential devices, including lighting, an air conditioner (AC) *(component of the HVAC system)*, and a motorized window blind controller. The lighting device features on/off control; the air conditioner allows users to set their desired temperature; and the motorized blinds can be controlled to open or close, effectively managing light exposure. From the selected devices, we chose specific activation phrases and control commands, as detailed in Table I. Like other voice control applications, each system typically requires at least one trigger word, commonly referred to as the *wake word* to activate the listening command mode. In this application, we selected **"Vừng ơi"** (Hey Vung) as the trigger phrase. This phrase is inspired by the story of "Ali Baba and the Forty Thieves" a popular fairy tale in Vietnam, where it serves as the command to open the treasure cave.

After activating the device with the wake word "Vừng ơi," users can issue corresponding commands to control the devices. The control commands are divided into two groups: single device control and group device control through scenario activation commands. To control the lighting, users can use the command pair **"bật đèn" - "tắt đèn"** (turn on light - turn off light). Similarly, the commands **"nóng quá" - "lạnh quá"** (too hot - too cold) adjust the temperature settings, while **"mở ra" - "đóng vào"** (open curtain - close curtain) manage the window blinds. Additionally, the commands **"xin chào" - "tạm biệt"** (hello - goodbye) control multiple devices when a user enters or leaves the room, streamlining communication.

B. Datasets Collecting

Based on the context presented in Section II-A, we collected an audio dataset that includes 9 command keywords, additional words, and noise sounds to facilitate the training of the model. The data was gathered from two sources: 1) a group of 20 students in the Rang Dong Lab at PTIT, and 2) some Text-to-Speech (TTS) platforms that support the Vietnamese language.

The data from the 20 students was recorded using Audacity software[1]. Each student was instructed to use Audacity to record the 9 keywords and some random additional words in mono mode. For the TTS platforms, we utilized the TTS services of three providers: Google[2], FPT-AI[3], and Viettel-AI[4]. To collect the data, we developed a Python program to connect with these platforms via their provided APIs. During data acquisition, we set the speech rate between 0.8 and 1.2, with a noise factor ranging from 0% to 20%. After processing, segmenting, and compiling the data, we obtained a dataset with a total duration of 4 hours and 23 minutes, with a detailed distribution presented in Table I.

C. Pre-Processing, Endpoint Detection & Features Extraction

The dataset, once compiled, was collected at various sampling frequencies. To ensure consistency and reduce computational demands suitable for implementation on MCU, all audio samples were downsampled to a sampling rate of 8,000 Hz. Subsequently, the signals were processed through a band pass filter (BPF) with a frequency range of 150 to 2,500 Hz, which aligns with the frequency range of human speech and effectively eliminates noise at other frequencies.

Although the audio was segmented to focus on speech, manual processing introduced some inconsistencies. To optimize the data for model training, we implemented an endpoint detection step [21], [22], which utilized an automated algorithm to retain only the segments containing speech. The endpoint detection algorithm employs both Short-Time Energy (STE) and Average Energy (AE) analyses to identify the locations of the audio segments. The values of STE and AE are calculated using formulas 1 and 2 [21], [22].

$$STE_k = \sum_{1}^{w} [y_k(i)]^2 \tag{1}$$

$$AE = \frac{\sum_{1}^{len} STE(i)}{len} \tag{2}$$

The identification of endpoints involves following key steps:

1) The STE is computed for each frame (utilizing 200 sample points for each frame).
2) A frame is marked as the beginning of a segment if its STE exceeds the upper threshold, STE_{high}.
3) Conversely, if the STE falls below the lower threshold (STE_{low}), for 5 consecutive frames, it is marked as the segment's end. If the STE

[1]https://www.audacityteam.org

[2]https://cloud.google.com/text-to-speech

[3]https://fpt.ai/vi/tts

[4]https://viettelai.vn/en

Table I
LIST OF WORDS USING ON THE EXPERIMENT

No.	Vietnamese	English	Role	Description	Samples
1.	**Vừng ơi**	hey Vung	wake word	**Prepare to listen control command**	1,475
2.	**đóng vào**	close	device control	Close the Curtain	1,496
3.	**mở ra**	open	device control	Open the Curtain	1,500
4.	**bật đèn**	turn on light	device control	Turn on the Light	1,450
5.	**tắt đèn**	turn off light	device control	Turn off the Light	1,429
6.	**nóng quá**	too hot	device control	Decrease set temperature of AC	1,453
7.	**lạnh quá**	too cold	device control	Increase set temperature of AC	1,537
8.	**xin chào**	hello	scene control	Turn on AC and Light	1,489
9.	**tạm biệt**	goodbye	scene control	Turn off AC and Light	1,472
10.	**-**	**"noise" and "others"**	**-**	**Noise and Unknown words/commands**	4,003
					17,304

is between these thresholds, the segment is extended.

4) After determining the endpoint, segments are merged if their duration (D) is within the specified range of $D1$ to $D2$; otherwise, they are discarded.
5) The AE of the entire segment is then calculated. Segments with an AE below the threshold, AE_{thre}, are discarded, while those meeting the threshold proceed to the feature extraction stage.

STE thresholds (STE_{low}, STE_{high}) were calculated using the following formula 3 [23].

$$STE_{thre} = STE_{min} + \alpha \times (\overline{STE} - STE_{min}) \quad (3)$$

where $\overline{STE}$ represents the mean STE value across all frames, STE_{min} denotes the minimum STE value among all frames, and the parameter α is assigned values of 0.1 and 0.4 for STE_{low} and STE_{high}, respectively. The thresholds for segment duration $D1$ and $D2$ were set to 20 and 40 frames, respectively. The outcome of the endpoint detection process is illustrated in Figure 1, where the blue line represents the raw audio signal, the green line depicts the STE values of segments, and the red line indicates the trimmed audio segments identified by this process. After completing the endpoint detection process, the total duration of the audio samples was reduced to 3 hours and 19 minutes with total 17,304 voice commands and unknown voices.

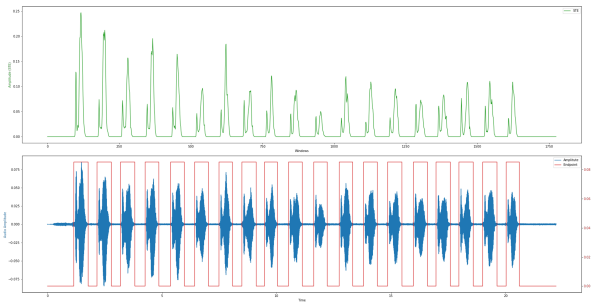

Figure 1. Vocalization endpoint detection employing techniques with the recording "Vừng ơi".

Mel-Frequency Cepstral Coefficients (MFCCs) [24], [25] are a crucial feature extraction technique widely utilized in speech and language recognition applications. MFCCs convert audio signals into a representation that closely aligns with human auditory perception by applying a Mel-scale filter bank, which effectively captures the phonetic characteristics of speech. This transformation emphasizes the frequency components most relevant to human hearing, enabling better discrimination of phonemes and improving the performance of machine learning models. In this study, MFCC will be employed to extract informative features from the audio data prior to inputting them into the machine learning model, ensuring that the model can effectively learn and recognize the nuances of the Vietnamese language.

D. Tiny Model Design

The design of the model is illustrated in Figure 2. The model's input consists of a matrix derived from the output of the MFCC block. Initially, the MFCC data is processed through 64 kernels of size 5×5. Subsequently, the data is further processed through 32 kernels of size 3×3. After two convolutional layers, the data is flattened and passed through two fully connected layers with 64 and 32 neurons, respectively. Finally, it reaches the neural classification layer, which has an output of 10, corresponding to the number of keywords the model recognizes. With these specifications, the model utilizes a total of 129,098

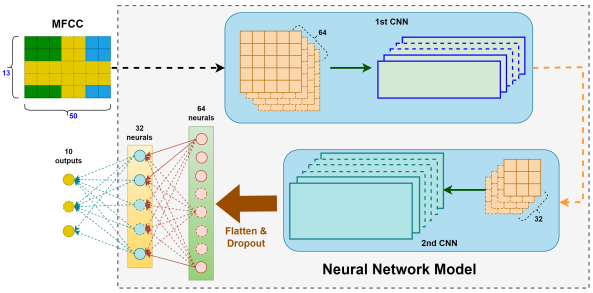

Figure 2. Design of neural model.

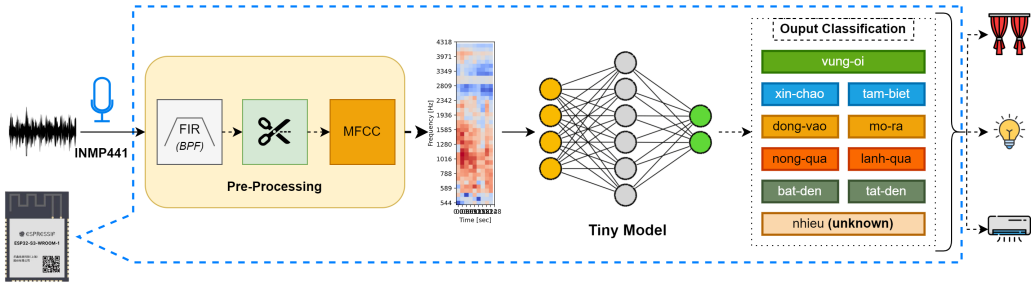

Figure 3. The architecture of firmware implementation on MCU.

parameters. The model was implemented, trained, and evaluated on the Edge Impulse platform[5], a powerful tool that supports the development of machine learning models for embedded applications. Edge Impulse allows developers to experiment with several popular machine learning models in the TinyML domain while providing essential DSP pre-processing blocks that streamline the experimentation and evaluation process. Once the construction, training, and evaluation are successfully completed, Edge Impulse supports the generation of programming libraries for MCU written in C/C++ with TensorFlow Lite Micro [26] integration, enabling developers to quickly deploy their models onto hardware.

E. Implementation

The model described in Section II-D, along with the MFCC computation block, was implemented with Edge Impulse support, resulting in a C/C++ codebase for MCU integration. This code served as the foundation for developing the firmware architecture for the KWS device, as shown in Figure 3. The firmware is designed for the ESP32-S3 microcontroller, which includes a FPU and basic DSP instructions, making it suitable for TinyML applications.

Environmental sound is captured by the INMP441 microphone sensor via I2S communication at a sampling rate of 8,000 Hz. The audio signal is filtered through a BPF with a range of 150 to 2,500 Hz and stored in a ring buffer that retains the last 5 seconds of audio. During data collection, every 200 samples trigger the calculation of STE for that segment. After each second of new data, the STE_{low} and STE_{high} values are recalculated, and the endpoint detection algorithm identifies newly occurring audio segments.

Once sufficient data is collected, MFCC extraction enables real-time keyword recognition, with the features fed into the trained machine learning model for audio classification. Based on these classifications, the MCU executes corresponding control commands. For monitoring purposes, 10 LEDs represent the recognized commands, illuminating for 2 seconds upon command identification, after which all LEDs automatically turn off. The device control implementation can be easily upgraded through peripheral expansion modules, allowing for flexible functionality enhancements.

III. RESULTS AND DISCUSSION

Following the pre-processing phase to extract suitable audio segments, a total of 17,304 files were uploaded to Edge Impulse for feature extraction and machine learning model training. Specifically, the dataset was divided into two main parts: 80% was allocated for training (with 60% used for training and 20% for validation), while the remaining 20% was designated for testing. The model configured as shown in the figure above was trained for 30 epochs with a learning rate of 0.005. After 30 epochs, the training accuracy exceeded 95%, while the corresponding loss decreased to below 0.15. The validation accuracy achieved approximately 94%, with a validation loss of around 0.23. The accuracy and loss curves for the training process are presented in Figure 4. Detailed information about the model and training results can be found in the Edge Impulse Studio[6].

The results of the testing with the test data are presented in the confusion matrix shown in Figure

[5]https://edgeimpulse.com/

[6]https://studio.edgeimpulse.com/studio/440801

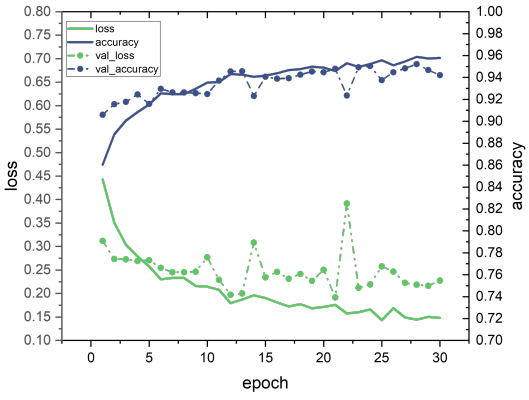

Figure 4. Accuracy and loss curves over training and validation.

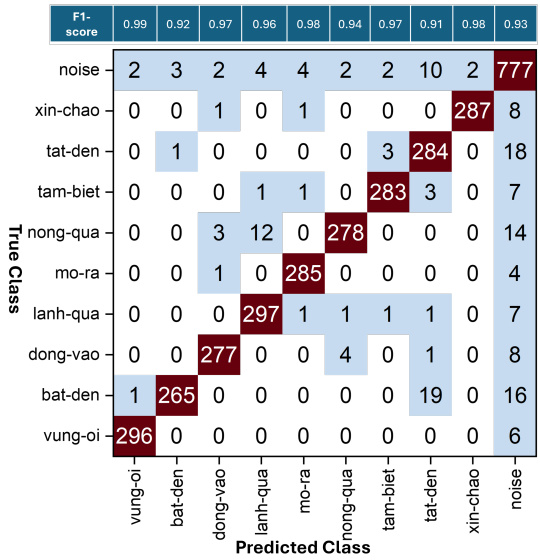

Figure 5. Confusion matrix for model evaluation on the test set with 8-bit quantization.

5. In terms of accuracy, most classes achieved over 90% accuracy, with the exception of the command "bat-den," which achieved an accuracy of 88%. The command "bat-den" was frequently confused with "tat-den", likely due to their strong acoustic resemblance. They share the word "đèn" and similar rhymes ("-ật"/"-ắt"), primarily differing in the initial consonant and tone, making them hard for the model to distinguish. The wake-up command "Vung-oi" demonstrated a particularly high accuracy of 98%. Most misclassifications were identified as confusion between control commands and noise, indicating that while the system performs well overall, there are challenges in distinguishing certain commands from background noise. Regarding overall performance metrics, the F1-score exceeded 0.91 for all classes, highlighting the model's robustness in balancing precision and recall across the different keywords.

The code for processing the selected trained model, generated by Edge Impulse, was integrated with the data acquisition, segmentation, and output control modules within the firmware development platform for the ESP32-S3 microcontroller. The experimental device, as depicted in Figure 6, was constructed and connected to upload the firmware. To evaluate the

Table II
EXECUTION TIME OF FIRMWARE ON ESP32-S3

No.	Step	Execution (ms)
1.	end-point detection	35
2.	DSP & MFCCs	276
3.	model inference	56
4.	control output peripherals	1
	Total	**368 ms**

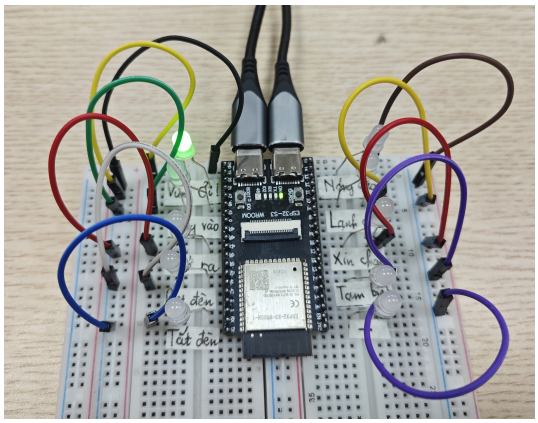

Figure 6. Implementation of prototype device.

real-time execution capabilities, the execution time for each stage of the firmware was measured, with results presented in Table II. The total execution time for each data window, which ranges from 500 ms to 1000 ms, was found to be 368 ms, indicating that the model is fully capable of real-time responses, thereby facilitating continuous data processing. Testing with audio signals demonstrated the device's capability to recognize control commands and execute the corresponding external controls as configured. However, it was observed that the device necessitates sound input at close distances and exhibits reduced performance in environments with significant background noise, such as those with music playing or operating machinery.

Compared to other research in KWS, our work offers distinct contributions, particularly when considering the application of TinyML for the Vietnamese language. Many existing TinyML KWS studies [15], [16], [17] have primarily focused on English or other widely spoken languages, often leveraging well-established datasets. Our research addresses an existing gap by developing and successfully implementing a KWS system specifically tailored for Vietnamese commands on a resource-constrained ESP32-S3 microcontroller.

A pertinent point of comparison within Vietnamese speech recognition is the work by Hung et al. [20]. Their research demonstrated Vietnamese speech command recognition using a BiLSTM model, which, while effective, typically requires more substantial computational resources, such as those available on a Raspberry Pi. In contrast, our system, utilizing a CNN model with 129,098 parameters, achieves an average accuracy of 94% and a real-time execution on the ESP32-S3. This performance on a low-cost MCU underscores the efficiency of our TinyML approach for localized voice control.

Our system's reliable real-time Vietnamese command recognition on resource-constrained MCUs represents a key contribution, especially given the limited

TinyML-based Vietnamese KWS studies available for direct benchmarking. This work establishes a practical pathway for accessible, offline, and privacy-preserving voice control for Vietnamese users. To further advance this nascent research area, future efforts will focus on implementing advanced DSP algorithms to improve environmental noise filtering, enhancing overall recognition accuracy, and conducting extensive practical testing within smart home systems.

IV. CONCLUSIONS

We demonstrated in this work that the integration of TinyML on the ESP32-S3 microcontroller can effectively achieve real-time keyword recognition. The proposed system showcased an overall accuracy exceeding 90% across most classes, with a notable performance in recognizing specific commands. Our results indicate that the model can process audio data within a total execution time of 368 ms for each data window, affirming its suitability for real-time applications. The results indicate significant potential for application in smart home systems, particularly in enhancing user experience for Vietnamese households.

ACKNOWLEDGMENT

This research was funded by the Posts and Telecommunications Institute of Technology (PTIT), Hanoi, Vietnam, grant number 03–2025–HV–KTĐT1.

REFERENCES

[1] S. Singh, S. Anand, and D. M. K. Satyarthi, "A Comprehensive Review of Smart Home Automation Systems," *Advances in Computer Science and Information Technology*, vol. 10, no. 2, 2023.

[2] S. N, S. S, and K. K N, "Voice Controlled Smart Home for Disabled," in *2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*. IEEE, 2024-01-24, pp. 1–4.

[3] M. Akour, "Mobile Voice Recognition Based for Smart Home Automation Control," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 3, pp. 3788–3792, 2020-06-25.

[4] S. Venkatraman, A. Overmars, and M. Thong, "Smart Home Automation—Use Cases of a Secure and Integrated Voice-Control System," *Systems*, vol. 9, no. 4, p. 77, 2021-10-28.

[5] A. H. Ruslan, A. Z. Jusoh, A. L. Asnawi, M. R. Othman, and N. I. Abdul Razak, "Development of multilanguage voice control for smart home with IoT," *Journal of Physics: Conference Series*, vol. 1921, no. 1, p. 012069, 2021-05-01.

[6] Y. Iliev and G. Ilieva, "A Framework for Smart Home System with Voice Control Using NLP Methods," *Electronics*, vol. 12, no. 1, p. 116, 2022-12-27.

[7] M. B. Hoy, "Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants," *Medical Reference Services Quarterly*, vol. 37, no. 1, pp. 81–88, 2018-01-02.

[8] S. Kennedy, H. Li, C. Wang, H. Liu, B. Wang, and W. Sun, "I Can Hear Your Alexa: Voice Command Fingerprinting on Smart Home Speakers," in *2019 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2019-06, pp. 232–240.

[9] M. A. Torad, B. Bouallegue, and A. M. Ahmed, "A voice controlled smart home automation system using artificial intelligent and internet of things," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 20, no. 4, p. 808, 2022-08-01.

[10] F. García-Vázquez, H. A. Guerrero-Osuna, G. Ornelas-Vargas, R. Carrasco-Navarro, L. F. Luque-Vega, and E. Lopez-Neri, "Design and Implementation of the E-Switch for a Smart Home," *Sensors*, vol. 21, no. 11, p. 3811, 2021-05-31.

[11] J. Bushur and C. Chen, "Neural Network Exploration for Keyword Spotting on Edge Devices," *Future Internet*, vol. 15, no. 6, p. 219, 2023-06-20.

[12] I. Lopez-Espejo, Z.-H. Tan, J. H. L. Hansen, and J. Jensen, "Deep Spoken Keyword Spotting: An Overview," *IEEE Access*, vol. 10, pp. 4169–4199, 2022.

[13] P. P. Ray, "A review on TinyML: State-of-the-art and prospects," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 4, pp. 1595–1623, 2022-04.

[14] V. Janapa Reddi, B. Plancher, S. Kennedy, L. Moroney, P. Warden, L. Suzuki, A. Agarwal, C. Banbury, M. Banzi, M. Bennett, B. Brown, S. Chitlangia, R. Ghosal, S. Grafman, R. Jaeger, S. Krishnan, M. Lam, D. Leiker, C. Mann, M. Mazumder, D. Pajak, D. Ramaprasad, J. E. Smith, M. Stewart, and D. Tingley, "Widening access to applied machine learning with TinyML," *Harvard Data Science Review*, 2022-01-27.

[15] S. Choi, S. Seo, B. Shin, H. Byun, M. Kersner, B. Kim, D. Kim, and S. Ha, "Temporal Convolution for Real-Time Keyword Spotting on Mobile Devices," in *Interspeech 2019*. ISCA, 2019-09-15, pp. 3372–3376.

[16] N. A. Abbas and M. R. Ahmad, "Keyword Spotting System With Nano 33 Ble Sense using Embedded Machine Learning Approach," *Jurnal Teknologi*, vol. 85, no. 3, pp. 175–182, 2023-04-19.

[17] I. López-Espejo, Z.-H. Tan, and J. Jensen, "An Experimental Study on Light Speech Features for Small-Footprint Keyword Spotting," in *IberSPEECH 2022*. ISCA, 2022-11-14, pp. 131–135.

[18] S. Garai and S. Samui, "Exploring TinyML frameworks for small-footprint keyword spotting: A concise overview," in *2024 International Conference on Signal Processing and Communications (SPCOM)*. IEEE, 2024-07-01, pp. 1–5.

[19] P. D. Hung, T. Minh, L. Hoang, and P. Minh, "Vietnamese Speech Command Recognition using Recurrent Neural Networks," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 7, 2019.

[20] P. D. Hung, T. M. Giang, L. H. Nam, P. M. Duong, H. Van Thang, and V. T. Diep, "Smarthome Control Unit Using Vietnamese Speech Command," in *Intelligent Computing and Optimization*, P. Vasant, I. Zelinka, and G.-W. Weber, Eds. Springer International Publishing, 2020, vol. 1072, pp. 290–300.

[21] K. Cuan, T. Zhang, J. Huang, C. Fang, and Y. Guan, "Detection of avian influenza-infected chickens based on a chicken sound convolutional neural network," vol. 178, p. 105688. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0168169920306621

[22] J. Huang, T. Zhang, K. Cuan, and C. Fang, "An intelligent method for detecting poultry eating behaviour based on vocalization signals," vol. 180, p. 105884. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0168169920330891

[23] K. Cuan, Z. Li, T. Zhang, and H. Qu, "Gender determination of domestic chicks based on vocalization signals," *Computers and Electronics in Agriculture*, vol. 199, p. 107172, Aug. 2022.

[24] J. Martinez, H. Perez, E. Escamilla, and M. M. Suzuki, "Speaker recognition using mel frequency cepstral coefficients (MFCC) and vector quantization (VQ) techniques," in *CONIELECOMP 2012, 22nd International Conference on Electrical Communications and Computers*. IEEE, 2012-02, pp. 248–251.

[25] Z. K. Abdul and A. K. Al-Talabani, "Mel frequency cepstral coefficient and its applications: A review," *IEEE Access*, vol. 10, pp. 122 136–122 158, 2022.

[26] R. David, J. Duke, A. Jain, V. J. Reddi, N. Jeffries, J. Li, N. Kreeger, I. Nappier, M. Natraj, S. Regev, R. Rhodes, T. Wang, and P. Warden, "TensorFlow lite micro: Embedded machine learning on TinyML systems," 2021-03-13. [Online]. Available: http://arxiv.org/abs/2010.08678

ỨNG DỤNG TINYML NHẬN DIỆN TỪ KHÓA TIẾNG VIỆT CHO ỨNG DỤNG ĐIỀU KHIỂN GIỌNG NÓI TRONG NHÀ THÔNG MINH

***Tóm tắt*—**Nghiên cứu này phát triển một hệ thống điều khiển giọng nói dựa trên TinyML dành riêng cho ngôn ngữ tiếng Việt, nhằm đáp ứng nhu cầu về các giải pháp nhà thông minh. Khác với các nghiên cứu hiện có tập trung vào tiếng Anh, nghiên cứu của chúng tôi tập trung vào đối tượng tiếng Việt. Chúng tôi đã thu thập một tập dữ liệu bao gồm một từ khóa đánh thức và tám lệnh điều khiển, tích hợp các thuật toán DSP và mô hình học máy trên vi điều khiển ESP32-S3. Mô hình này có 129,098 tham số, đạt độ chính xác trung bình 94% và thời gian thực hiện thời gian thực là 368 ms cho các cửa sổ dữ liệu từ 500-1,000 ms. Điều này cho thấy tiềm năng của TinyML trong các thiết bị điều khiển bằng giọng nói, cung cấp một giải pháp tiết kiệm chi phí, độc lập với Internet, đồng thời nâng cao bảo mật dữ liệu và quyền riêng tư của người dùng.

***Từ khóa*—**Nhà thông minh, Nhận diện từ khóa, Điều khiển giọng nói, Phát hiện đoạn âm thanh, Học máy nhỏ, TinyML, MFCCs.

Duan Luong Cong received a BE degree in Electrical - Electronic Engineering in 2014 and an MSc in Telecommunication Engineering in 2018 from the Posts and Telecommunications Institute of Technology (PTIT). He is a lecturer in the Faculty of Electronic Engineering 1 at PTIT. His main research areas include Embedded Systems, DSP, Internet of Things (IoT), and TinyML.
Email: duanlc@ptit.edu.vn

Cuong Chu Van received a BE degree in Electrical - Electronic Engineering in 2021 and an MSc in Electronic Engineering in 2024 from the Posts and Telecommunications Institute of Technology (PTIT). He is a lecturer in the Faculty of Electronic Engineering 1 at PTIT. His main research areas include Embedded Systems, Internet of Things (IoT), and Robotics.
Email: cuongcv@ptit.edu.vn

Minh Nguyen Ngoc graduated with a Ph.D. in Electrical Engineering from La Trobe University in 2007. He is a lecturer in the Faculty of Electronic Engineering 1 at the PTIT. His main research areas include Embedded Systems, FPGA, and DSP.
Email: minhnn@ptit.edu.vn