JTE HCMUTE
ISSN: 1859-1272

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.hcmute.edu.vn/index.php/jte/index
Email: jte@hcmute.edu.vn

Realtime Non-invasive Fault Diagnosis of Three-phase Induction Motor

Nguyen Van Khanh[1*], Tran Vy Khang[2], Nguyen Minh Khai[2], Thach Van To Em[2], Pham Tran Lam Hai[1], Nguyen Chi Ngon[1]

[1]*Can Tho University, Vietnam*
[2]*Automation and Control Engineering, Can Tho University, Vietnam*

[*] Corresponding author. *Email: vankhanh@ctu.edu.vn*

ARTICLE INFO

Received: 20/06/2022
Revised: 14/08/2022
Accepted: 06/10/2022
Published: 28/10/2022

KEYWORDS

Non-invasive fault diagnosis;
Spectrogram;
esp32;
deep learning network;
embedded system.

ABSTRACT

The objective of this paper is to apply deep learning network running on an embedded system platform to diagnose faults of a three-phase electric motor by a non-contact method based on operating motor noise. To accomplish this, at first, deep learning network should be designed and trained on a computer, and then converted to an equivalent network to run on the embedded system. The network input data is a two-dimension spectrogram image of the noise emitted by the motor in four main cases, including normal operation, phase shift, phase loss and bearing failure. The execution time and accuracy of these deep learning network structures will be deployed on three microcontrollers including ESP32, ESP32-C3 and nRF52840 to determine the suitable embedded platform and network structure for real-time running. Experimental results show that the proposed deep learning network models could diagnose the faults well on both computer and embedded platform with the highest accuracies are 99,7% and 99,3%, respectively. In particular, the preliminary results are remarkable with the recognition time and accuracy at 1,7 seconds and 72%, respectively associated with the proposed deep learning network on realtime embedded system performance.

Chẩn Đoán Thời Gian Thực Không Xâm Lấn Lỗi Động Cơ Điện Ba Pha

Nguyễn Văn Khanh[1*], Trần Vỹ Khang[2], Nguyễn Minh Khải[2], Thạch Văn Tố Em[2], Phạm Trần Lam Hải[1], Nguyễn Chí Ngôn[1]

[1]*Đại học Cần Thơ, Việt Nam*
[2]*Kỹ thuật điều khiển và Tự động hóa, Đại học Cần Thơ, Việt Nam*

** Tác giả liên hệ. Email: vankhanh@ctu.edu.vn*

THÔNG TIN BÀI BÁO

Ngày nhận bài: 20/06/2022
Ngày hoàn thiện: 14/08/2022
Ngày chấp nhận đăng: 06/10/2022
Ngày đăng: 28/10/2022

TỪ KHÓA

Chẩn đoán lỗi không tiếp xúc;
ảnh phổ tần số;
esp32;
mạng học sâu;
hệ thống nhúng.

TÓM TẮT

Mục tiêu của bài báo là ứng dụng mạng học sâu chạy trên nền tảng hệ thống nhúng để chẩn đoán thời gian thực lỗi động cơ điện ba pha bằng phương pháp không tiếp xúc dựa trên tiếng ồn phát ra. Để thực hiện điều này, trước tiên, các mạng học sâu cần được thiết kế và huấn luyện trên máy tính trước khi được chuyển đổi thành mạng tương đương phù hợp với hệ thống nhúng. Ngõ vào của mạng là ảnh phổ hai chiều của tiếng ồn phát ra từ động cơ trong bốn trường hợp: bình thường, lệch pha, mất pha và vỡ bạc đạn. Thời gian thực thi và độ chính xác của các cấu trúc mạng học sâu sẽ được khảo sát trên ba vi điều khiển là ESP32, ESP32-C3 và nRF52840 để chọn ra vi điều khiển và kiến trúc mạng phù hợp nhất để chạy thời gian thực. Kết quả thực nghiệm cho thấy các kiến trúc mạng học sâu đề nghị đã chẩn đoán tốt các lỗi động cơ trên cả hai nền tảng máy tính và hệ thống nhúng với độ chính xác cao nhất tương ứng là 99,7% và 99,3%. Đặc biệt khi chạy thời gian thực kiến trúc mạng đã chọn trên hệ thống nhúng cũng đã cho kết quả ban đầu rất ấn tượng với thời gian nhận dạng và độ chính xác tương ứng là 1,7 giây và 72%.

Doi: https://doi.org/10.54644/jte.72B.2022.1231

Copyright © JTE. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International License which permits unrestricted use, distribution, and reproduction in any medium for non-commercial purpose, provided the original work is properly cited.

1. Giới thiệu

Trong những năm gần đây, rất nhiều nghiên cứu tập trung vào các giải pháp để chẩn đoán lỗi động cơ điện đang vận hành dựa trên phân tích các tín hiệu âm thanh, rung động hay dòng điện để dự báo sớm các hư hỏng có thể xảy ra nhằm giảm thiểu các rủi ro trong vận hành hệ thống [1]. Bên cạnh đó, mạng nơ-ron nhân tạo (AI – Artificial intelligence) cũng đã và đang được áp dụng vào lĩnh vực này để tạo ra một phương pháp chẩn đoán thông minh nhằm nâng cao độ tin cậy và đảm bảo các hệ thống kỹ thuật hoạt động an toàn và liên tục [2, 3]. Việc dự đoán các hoạt động bảo dưỡng và sửa chữa dựa trên kết quả chẩn đoán đã được áp dụng phổ biến và đem lại lợi ích về kinh tế, tính năng này đã MathWorks tích hợp thành một Toolbox trên phần mềm Matlabs nổi tiếng của họ [4]. Thực thi các mạng học sâu trên hệ thống nhúng hay thiết bị di động là một bước đi rất cần thiết để triển khai chúng vào ứng dụng thực tiễn.

Việc nghiên cứu và ứng dụng mô hình máy học quy mô nhỏ (hay còn được gọi là TinyML) đã được nhiều nhà nghiên cứu quan tâm. TinyML đề cập đến hướng nghiên cứu về học máy để tối ưu, nén các mô hình AI sao cho có thể chạy được trên các MCU (Microcontroller unit) hạn chế về tài nguyên tính toán và lưu trữ [5]. Trong bài viết của mình tác giả Ray đã trình bày rất chi tiết về khả năng của TinyML và đánh giá về nó một cách trực quan [6]. Một số dự án áp dụng TinyML đã được triển khai trên nhiều lĩnh vực, điển hình như thiết bị đeo theo dõi sức khỏe là giải pháp hiệu quả để biết được tình trạng hiện tại của người sử dụng nhằm đưa ra các cảnh báo và lời khuyên phù hợp [7]. Mô hình lái xe tự hành của tác giả Prado cho thấy được các thách thức về khả năng tính toán và lưu trữ đối với mạng học sâu chạy trên hệ thống nhúng [8]. Các nghiên cứu này cho thấy khả năng ứng dụng TinyML trong chẩn đoán thời gian thực lỗi động cơ là hoàn toàn khả thi. Tuy nhiên, một câu hỏi lớn đặt ra là cấu trúc mạng học sâu và MCU nào là phù hợp nhất để thực hiện ứng dụng chạy thời gian thực.

Mục tiêu của bài báo này là đề xuất một kiến trúc mạng học sâu TinyML phù hợp để chẩn đoán lỗi động cơ điện ba pha dựa trên tiếng ồn phát ra khi vận hành. Một số kiến trúc mạng học sâu chẩn đoán lỗi động cơ sẽ được đề xuất, huấn luyện và thử nghiệm để chọn ra một kiến trúc phù hợp nhất. Khả năng thực thi mạng cũng được thử nghiệm trên một số dòng MCU thông dụng hiện nay để giúp người sử dụng có thể đánh giá và lựa chọn một MCU hợp lý cho ứng dụng của họ. Mô hình mạng tối ưu sẽ được cài đặt và thực nghiệm thời gian thực trên MCU được chọn. Thành công của nghiên cứu này sẽ mở ra nhiều hướng nghiên cứu liên quan đến nhận dạng tín hiệu thời gian thực.

2. Dữ liệu và phương pháp

2.1 Dữ liệu

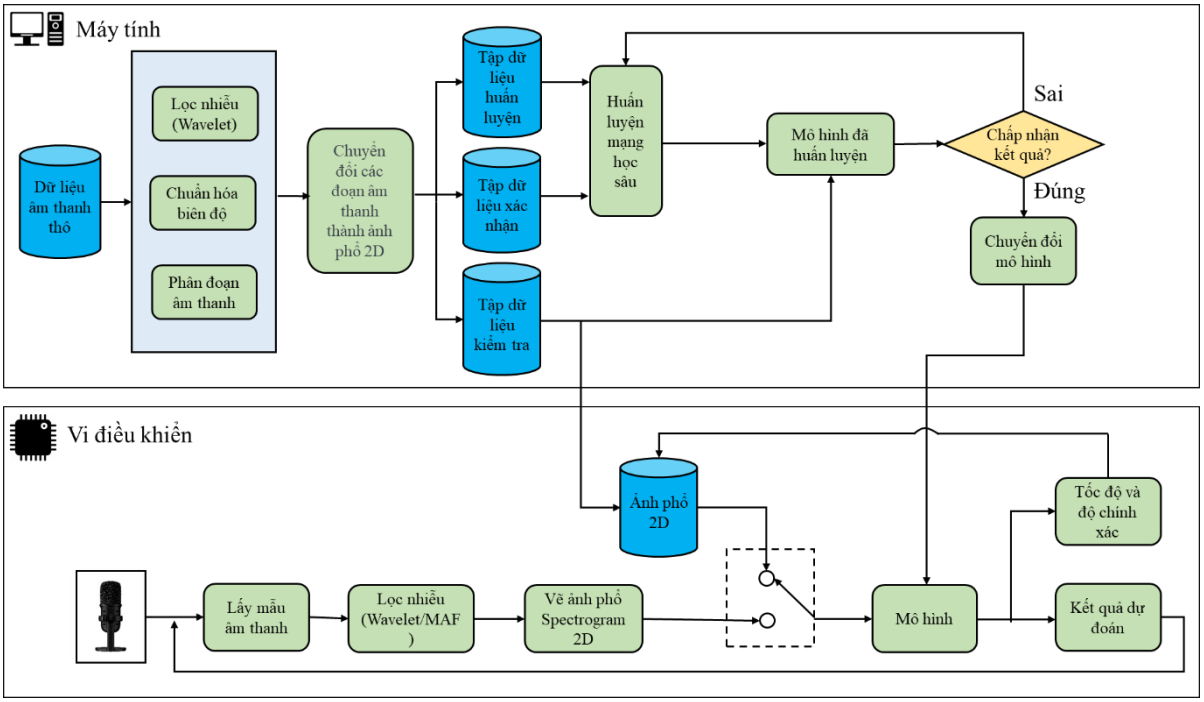
Dữ liệu huấn luyện là ảnh phổ tần số 2D (Two-dimension) của bốn loại âm thanh phát ra từ một động cơ điện xoay chiều ba pha tương ứng với trường hợp động cơ hoạt động bình thường, bị lệch pha, mất pha và vỡ bạc đạn. Thông tin về thông số động cơ, bố trí thí nghiệm thu âm được trình bày trong nghiên cứu [3], một công bố khác trong chuỗi nghiên cứu của nhóm tác giả. Nghiên cứu này sử dụng lại tập dữ liệu âm thanh, tuy nhiên quá trình xử lý tín hiệu, tạo ảnh phổ được viết bằng ngôn ngữ Python và thực hiện hoàn toàn trên công cụ Colab [9] hãng Google thay vì sử dụng ngôn ngữ Matlabs. Dữ liệu âm thanh của các nhóm khi được chuyển đổi thành ảnh phổ sẽ được gán nhãn, phân chia thành ba tập dữ liệu chính bao gồm *Train, Validation* và *Test* theo tỉ lệ tương ứng là 60%, 20% và 20% để sử dụng cho huấn luyện và kiểm tra mạng học sâu.

2.2 Phương pháp

Hình 1 trình bày tổng quan về phương pháp thực hiện của nghiên cứu này bao gồm hai giai đoạn chính thực hiện trên máy tính và trên MCU. Trên máy tính, các mô hình mạng học sâu sau khi thiết kế sẽ được xây dựng và huấn luyện trên công cụ Colab sử dụng tập ảnh phổ tần số 2D của tiếng ồn thu từ động cơ. Sau khi quá trình huấn luyện kết thúc, mạng học sâu sẽ được kiểm tra lại với tập dữ liệu *Test* để đánh giá xem nó có đạt yêu cầu không, nếu không, thông số hay cấu trúc mạng sẽ được hiệu chỉnh để huấn luyện lại cho đến khi đạt yêu cầu. Mạng học sâu sau khi huấn luyện thành công sẽ được chuyển đổi thành mô hình mạng có thể hoạt động trên nền tảng MCU.

JTE HCMUTE
ISSN: 1859-1272
JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.hcmute.edu.vn/index.php/jte/index
Email: jte@hcmute.edu.vn

Giai đoạn trên MCU sẽ thực hiện hai tác vụ quan trọng. Thứ nhất, kiểm tra tốc độ thực thi mạng khi nhận dạng các ảnh phổ trong tập dữ liệu *Test* trên máy tính để chọn ra kiến trúc mạng cũng như MCU phù hợp. Thứ hai, mạng học sâu tối ưu nhất sẽ được chạy nhận dạng thời gian thực với nguồn âm thanh thu trực tiếp từ micro tích hợp. Để thực hiện được điều này, MCU sẽ thu các đoạn âm thanh có thời lượng một giây, lọc nhiễu, chuẩn hóa biên độ và chạy thuật toán tạo ảnh phổ tần số 2D trước khi đưa vào mạng học sâu để nhận dạng hay để đánh giá tình trạng của động cơ.



Hình 1. *Tổng quan về phương pháp thực hiện*

2.2.1 Thuật toán xử lý tín hiệu âm thanh

Trong bài báo này, bộ lọc Wavelet và lọc trung bình dịch chuyển (MAF – Moving average filter) sẽ được áp dụng để lọc nhiễu tín hiệu âm thanh. Trên máy tính, dữ liệu âm thanh sau khi thu thập sẽ được chuẩn hóa biên độ về trong khoảng [-1; 1], loại bỏ các khoảng lặng nếu có ở đầu và cuối đoạn âm thanh thu được (xuất hiện do thủ tục thu dữ liệu thủ công) và lọc nhiễu bằng bộ lọc Wavelet.

Bộ lọc Wavelet được áp dụng trong cả hai trường hợp xử lý tín hiệu trên máy tính và MCU. Bộ lọc này có thuật toán được trình bày khá chi tiết trong [10]. Tín hiệu âm thanh sẽ được xử lý qua ba bước chính bao gồm 1) phân giải tín hiệu thành các hệ số Wavelet w_i sử dụng DWT (Discrete wavelet transform), 2) xác định ngưỡng để loại bỏ nhiễu và 3) phục hồi tín hiệu sau khi loại bỏ nhiễu bằng biến đổi IDWT (Inverse DWT). Nghiên cứu này sử dụng ngưỡng Wavelet mềm (Soft threshold) được minh họa toán học bằng công thức (1).

$$D_s = \begin{cases} sgn(w)(|w| - T) & nếu\ |w| \geq T \\ 0 & nếu\ |w| < T \end{cases} \quad (1)$$

Trong đó, D_s là các hệ số Wavelet sau khi áp dụng ngưỡng mềm, w là các hệ số Wavelet của tín hiệu gốc và T là giá trị của ngưỡng.

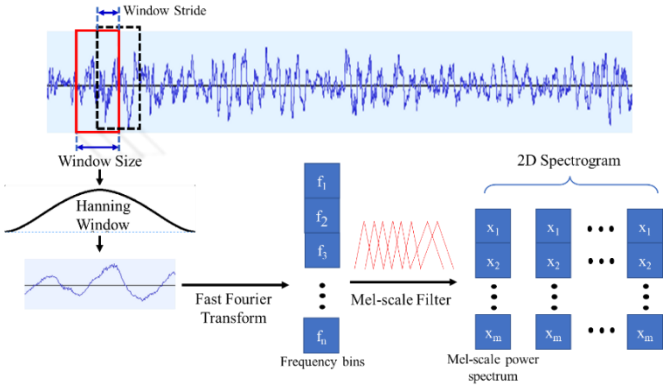
Bên cạnh bộ lọc Wavelet thì bộ lọc MAF cũng được áp dụng để lọc tín hiệu âm thanh khi chạy thời gian thực trên MCU để tăng tốc độ thực thi. MAF là một bộ lọc khá đơn giản có biểu diễn toán học như công thức (2) [11]. Tham số chính của MAF là số phần tử hay kích thước M của cửa sổ trượt. Cửa sổ này sẽ trượt từ đầu đến cuối tín hiệu mỗi lần dịch chuyển một mẫu, tại mỗi vị trí của cửa sổ sẽ tính toán được giá trị của tín hiệu ngõ ra chính là trung bình của các phần tử trong cửa sổ. MAF rất hiệu quả trong làm trơn tín hiệu, loại các nhiễu tần số cao, nhưng tín hiệu ra bị lệch pha so với tín hiệu vào.

JTE
HCMUTE
ISSN: 1859-1272

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.hcmute.edu.vn/index.php/jte/index
Email: jte@hcmute.edu.vn

$$y(n) = \frac{1}{M}\sum_{i=0}^{M-1} x(n+i) \quad (2)$$

Trong đó, *y(n)* là tín hiệu ngõ ra sau khi lọc, *x* là tín hiệu ngõ vào và *M* là kích thước của cửa sổ. Trong bài báo này, tín hiệu ngõ vào chính là âm thanh mà vi điều khiển thu được từ micro, ngõ ra sẽ được sử dụng để tạo ảnh phổ tần số.



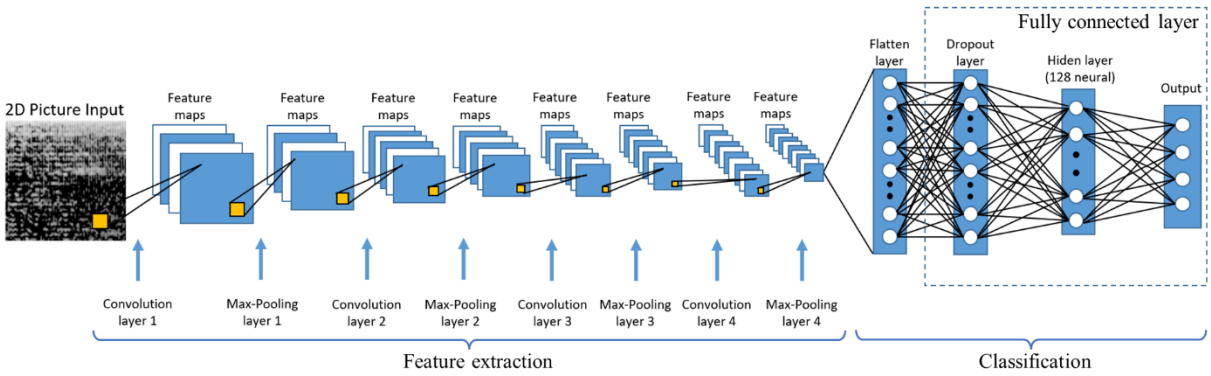
Hình 2. *Nguyên lí tạo ảnh phổ tần 2D từ tín hiệu âm thanh*

2.2.2 Thuật toán tạo ảnh phổ

Tín hiệu sau khi lọc sẽ được phân chia thành các đoạn nhỏ có thời lượng một giây đều nhau để chuyển đổi thành ảnh phổ 2D sử dụng thuật toán như được trình bày trong *Hình 2*. Nguyên lý tổng quát đó là một cửa sổ có kích thước là *window size* sẽ trượt từ đầu đến cuối tín hiệu. Các cửa sổ này sẽ chồng lên nhau một khoảng *window stride*. Tại mỗi vị trí của cửa sổ tín hiệu sẽ được lấy mẫu bằng cửa sổ Hann [12] và được chuyển đổi FFT (Fast Fourier Transform) bằng thuật toán Kiss-FFT [13] để thu được *n=256* giá trị biên độ của các thành phần tần số. Tùy thuộc vào kích thước ảnh phổ cần tạo, 256 giá trị này sẽ được nhóm lại và lấy trung bình bằng hàm Mel [14], số lượng nhóm tần số bằng với số cột của ảnh phổ tần số ngõ ra. Trong nghiên cứu này có ba kích thước ảnh phổ được sử dụng đó là 48x48, 64x64 và 96x96 pixel. Các ảnh phổ này sẽ được sử dụng làm dữ liệu để huấn luyện và kiểm tra mạng học sâu.

2.2.3 Cấu trúc mạng học sâu đề nghị

Mạng học sâu đề nghị trong nghiên cứu này có kiến trúc tổng quát như được trình bày trong *Hình 3*, được hiệu chỉnh dựa trên kiến trúc được áp dụng trong [15, 16]. Tổng quát mạng học sâu bao gồm hai phần chính là trích xuất đặc trưng của ngõ vào (Feature extraction) và phân loại ảnh ngõ vào (Classification). *Lớp trích đặc trưng* được tạo nên từ 4 lớp *Tích chập 2D* để trích xuất các đặc trưng của ảnh ngõ vào thành các bản đồ đặc trưng (Feature map), sau mỗi lớp *tích chập* sẽ là một lớp *Max-pooling* để giảm kích thước của *Feature map* và tăng tốc quá trình huấn luyện cũng như giảm tham số mạng.



Hình 3. *Kiến trúc mạng học sâu được đề nghị*

Cấu trúc này đã được áp dụng trong [16] để phân loại các âm thanh của động cơ điện cho kết quả rất khả quan. Lớp *phân loại* gồm có một lớp *Flatten* để làm phẳng các *feature map* từ hai chiều sang một chiều duy nhất. Lớp *Dropout* được sử dụng để loại bỏ ngẫu nhiên một số nút mạng trong quá trình huấn luyện nhằm giảm thiểu hiện tượng *overfiting* [17]. Tiếp theo là một lớp *Hiden layer* gồm 128 nơ-ron sử dụng hàm kích hoạt ReLU kết nối với ngõ ra sử dụng hàm kích hoạt *Softmax* cho 4 nơ-ron trả về các giá trị dữ đoán của bốn loại âm thanh dưới dạng xác xuất trong khoảng [0;1].

Bảng 1. *Thông số các cấu trúc mô hình học sâu được đề nghị*

Kiến trúc	Convolution layer 1 Feature maps	Convolution layer 2 Feature maps	Convolution layer 3 Feature maps	Convolution layer 4 Feature maps	Convolution layer 1 Kernel size	Convolution layer 2, 3, 4 Kernel size	Global Max-Pooling Kernel size
1	32	64	128	128	5x5	3x3	2x2
2	16	32	64	128	5x5	3x3	2x2
3	8	16	32	64	5x5	3x3	2x2

Số *feature map* của từng lớp *tích chập 2D* sẽ được thay đổi để tạo ra ba kiến trúc mạng học sâu khác nhau, Bảng 1 trình bày tóm tắt các tham số của chúng. Kiến trúc 1 có số *feature map* lớn nhất ở mỗi lớp *tích chập 2D* và giảm dần đến kiến trúc 3, các thông số còn lại là tương tự nhau ở mỗi kiến trúc. Mỗi kiến trúc sẽ có ba kích thước ngõ vào khác nhau là 48x48, 64x64 và 96x96 tạo nên tổng cộng 9 kiến trúc khác nhau tương ứng với số lượng tham số huấn luyện được trình bày trong Bảng 2.

Bảng 2. *Số lượng tham số huấn luyện của các kiến trúc mạng học sâu đã đề nghị*

Kiến trúc	Kích thước ảnh ngõ vào		
	48x48	64x64	96x96
1	257796	306948	503556
2	114436	163588	360196
3	33348	57924	156228

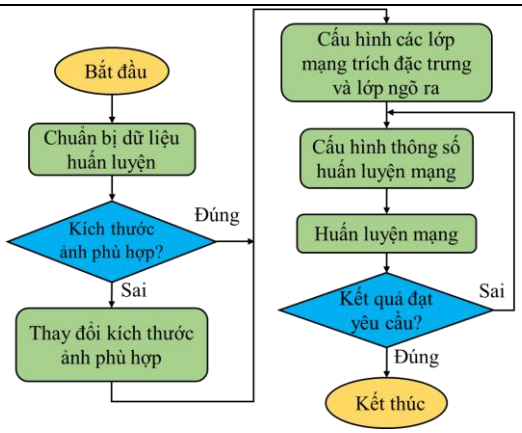
2.2.4 Huấn luyện và chuyển đổi mạng học sâu để thực thi trên MCU

Các mô hình mạng học sâu sau khi thiết kế sẽ được cài đặt và huấn luyện bằng ngôn ngữ Python trên công cụ Colab của hãng Google. Quá trình này được trình bày trong lưu đồ thuật toán ở *Hình 4* gồm hai công đoạn chính. Thứ nhất là chuẩn bị, gán nhãn và phân loại dữ liệu sẵn sàng cho huấn luyện. Thứ hai là cài đặt mô hình mạng dựa trên nền tảng *Tensorflow* [18], sau đó sẽ thực thi vòng lập huấn luyện với dữ liệu đã chuẩn bị. Mạng học sâu sau khi được huấn luyện được kiểm tra với tập dữ liệu *Test*, nếu chưa đạt yêu cầu, các tham số của cấu trúc mạng sẽ được hiệu chỉnh và huấn luyện lại. Ngược lại, nếu mạng đã đạt yêu cầu thì lập lại toàn bộ quá trình với các mạng còn lại. Đối với các mạng sử dụng chung kích thước ảnh ngõ vào thì chỉ cần thay đổi tham số mạng trước khi huấn luyện lại. Tổng cộng có 9 mạng học sâu như liệt kê trong Bảng 2 sẽ được huấn luyện trong nghiên cứu này.

Các mạng học sâu được cài đặt bằng *Tensorflow* sau khi được huấn luyện sẽ được biểu diễn dưới dạng nhị phân của các số dấu chấm động dạng chính xác đơn *Binary32* theo tiêu chuẩn IEEE 754. Với biểu diễn này, các mạng học sâu sau khi huấn luyện có kích thước rất lớn không phù hợp để thực thi trên các MCU có tài nguyên hạn chế. Vì vậy, chúng cần được lượng tử hóa thành các biểu diễn khác để có thể triển khai trên MCU. Nghiên cứu này sử dụng phương pháp lượng tử hóa sau khi huấn luyện dựa trên công cụ *Tensorflow Lite* [19] vì nó dễ thực hiện, kích thước mô hình sau lượng tử giảm đáng kể và độ chính xác không chênh lệch quá lớn [20, 21].

JTE
HCMUTE
ISSN: 1859-1272

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.hcmute.edu.vn/index.php/jte/index
Email: jte@hcmute.edu.vn



Hình 4. *Thuật toán quá trình huấn luyện mạng học sâu trên máy tính*

2.2.5 Đánh giá tốc độ thực thi mạng trên MCU

Vấn đề đặt ra sau khi lượng tử hóa các mạng học sâu sau huấn luyện đó là MCU và kiến trúc mạng nào phù hợp để chạy thời gian thực với tốc độ nhận dạng và độ chính xác tối ưu nhất? Để giải đáp câu hỏi này, ba dòng MCU khá phổ biến hiện nay là nRF52840 được tích hợp trên kit Arduino Nano 33 BLE Sense, ESP32 được tích hợp trên kit Dev-Kit V1 và ESP32-C3 được tích hợp trên Kit NodeMCU-C3-32S Ai-Thinker sẽ được sử dụng để chạy thử nghiệm tất cả các mạng học sâu sau lượng tử. Trong các kit hỗ trên thì Arduino Nano 33 BLE Sense là phù hợp nhất để chạy thời gian thực mạng học sâu vì kit này tích hợp sẵn micro thu âm.

Tốc độ và độ chính xác của các mạng học sâu chạy trên các MCU sẽ được kiểm tra bằng cách áp dụng chúng để nhận diện các ảnh phổ trong tập dữ liệu *Test*. Các ảnh này cần được chuyển đổi thành mảng hai chiều kiểu 8 bit (*int8*) để phù hợp với mạng sau lượng tử. Mỗi MCU sẽ được kiểm tra với 400 ảnh phổ cho mỗi loại âm thanh thu được từ động cơ. Độ chính xác và *F1-score* tương ứng của từng mạng học sâu được sử dụng để làm cơ sở đánh giá chọn MCU và mạng phù hợp nhất để chạy nhận dạng thời gian thực.

2.2.6 Các chỉ tiêu đánh giá kết quả

Sau khi huấn luyện, giá trị độ chính xác (Accuaracy) và chỉ số *F1-score* được sử dụng để đánh giá cũng như so sánh các cấu trúc mạng với nhau. Độ chính xác được tính bởi công thức (3) như sau:

$$\text{Accuracy} = \frac{\sum TP + TN}{\sum TP + FP + FN + TN} \tag{3}$$

Trong đó, *Accuracy* là độ chính xác của mạng cần đánh giá, *TP* (True Positive) là số lượng dự đoán chính xác, *TN* (True Negative) là số lương dự đoán chính xác một cách gián tiếp, *FP* (False Positive) là số lượng các dự đoán sai lệch, *FN* (False Negative) là số lượng các dự đoán sai lệch một cách gián tiếp.

F1-score được tính bằng công thức (4) như sau:

$$F1 - score = \frac{2 * Recall * Precision}{Recall + Precision} \tag{4}$$

Với:

$$Precision = \frac{TP}{TP + FP}, \tag{5}$$

$$Recall = \frac{TP}{TP + FN}. \tag{6}$$

Một cấu trúc mạng có chỉ số *F1-score* cao khi và chỉ khi cả hai chỉ số *Precision* và *Recall* đều cao. *F1-score* sẽ thấp khi một trong hai chỉ số này có giá trị thấp. Trường hợp xấu nhất là khi một trong hai chỉ số *Precison* và *Recall* bằng 0 thì *F1-score* bằng 0. Để *F1-score* bằng 1 thì cả hai giá trị này phải bằng 1. Do đó chỉ số *F1-score* là một thước đo đáng tin cậy về hiệu năng của mạng học sâu trong các bài toán phân loại.

3. Kết quả và thảo luận

3.1 Khả năng nhận dạng lỗi động cơ của các mạng học sâu đã đề xuất

Kết quả huấn luyện các mạng học sâu trên máy tính được thể hiện trong Bảng 3. Kết quả này cho thấy độ chính xác tổng thể của các mạng học sâu sử dụng kiến trúc 1 là cao nhất và giảm dần đến kiến trúc 3. Điều này khá dễ hiểu vì kiến trúc 1 có quy mô lớn nhất và giảm dần đến trúc 3. Tương tự, kích thước ảnh phổ cũng ảnh hưởng đến độ chính xác huấn luyện, ảnh 96x96 pixel cho mạng có độ chính xác cao nhất và giảm dần đến ảnh có kích thước 48x48.

Bảng 3. *Kết quả huấn luyện các mạng học sâu trên máy tính*

Kiến trúc	Kích thước ảnh	Độ chính xác	F1-score của bình thường	F1-score của lệch pha	F1-score của mất pha	F1-score của vỡ bạc đạn	Thời gian huấn luyện
1	48x48	97,59%	97,27%	95,67%	97,68%	99,75%	75 phút
	64x64	99,63%	99,75%	99,30%	99,45%	100%	68 phút
	96x96	**99,66%**	**99,69%**	**99,37%**	**99,56%**	**100%**	**58 phút**
2	48x48	96,97%	97,13%	94,44%	96,64%	99,69%	75 phút
	64x64	99,09%	99,31%	98,19%	98,95%	99,94%	68 phút
	96x96	**99,57%**	**99,75%**	**99,12%**	**99,38%**	**100%**	**63 phút**
3	48x48	94,96%	95,39%	90,70%	91,34%	99,43%	92 phút
	64x64	98,78%	99,06%	97,63%	98,49%	99,94%	78 phút
	96x96	**98,94%**	**98,95%**	**97,86%**	**98,93%**	**100%**	**69 phút**

Bên cạnh độ chính xác, *F1-score* cũng được sử dụng để đo độ chính xác của mạng đối với từng loại âm thanh riêng lẻ. Điểm chung của các kiến trúc mạng học sâu đó là nhận dạng âm thanh phát ra do hiện tượng vỡ bạc đạn có độ chính xác cao nhất. Hai trường hợp âm thanh của hiện tượng mất pha và lệch pha dễ xảy ra nhầm lẫn ở tất cả các kiến trúc mạng. Đối với âm thanh phát ra khi động cơ hoạt động bình thường các mạng nhận dạng khá chính xác nhưng vẫn thấp hơn khi nhận dạng âm thanh vỡ bạc đạn.

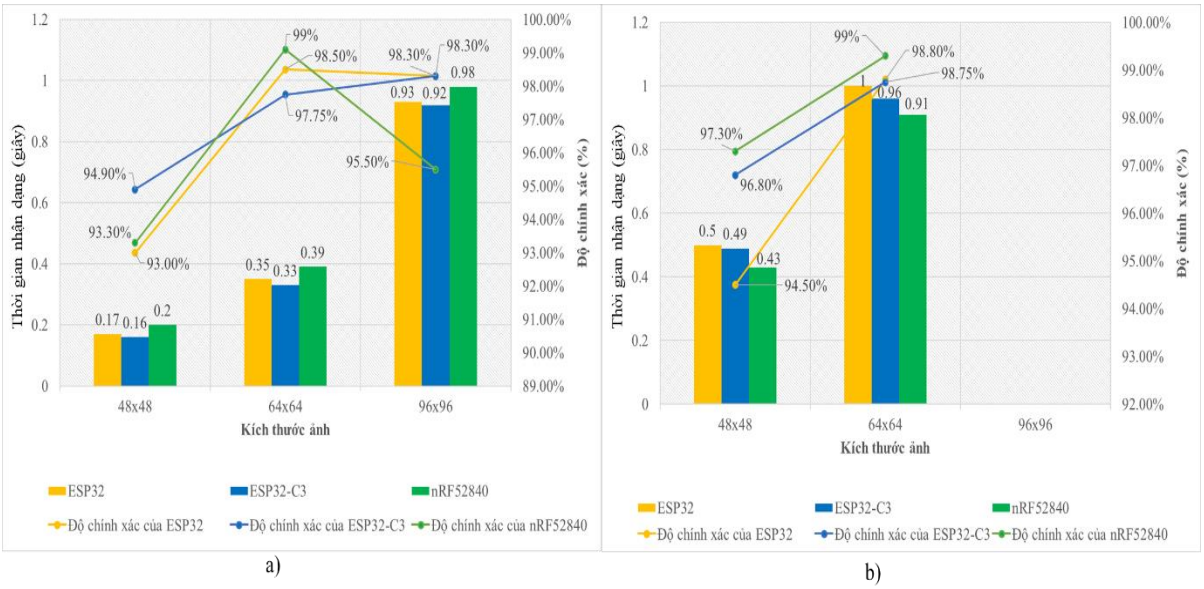
Tóm lại, mạng học sâu sử dụng kiến trúc 1 với ảnh ngõ vào 96x96 *pixel* cho độ chính xác nhận dạng tổng thể cũng như từng loại âm thanh cao nhất cùng với thời gian huấn luyện thấp nhất. Ngược lại, mạng học sâu sử dụng kiến trúc 3 với ảnh ngõ vào 48x48 *pixel* cho có độ chính xác nhận dạng tổng thể cũng như từng loại âm thanh thấp nhất cùng với thời gian huấn luyện dài nhất. Tất cả các mạng học sâu sau khi huấn luyện sẽ được lượng tử hóa và thực thi trên MCU để tìm ra một kiến trúc mạng phù hợp nhất, vừa có thời gian thực thi nhanh vừa đạt độ chính xác cao.

3.2 Khả năng thực thi các mạng học sâu trên các MCU

Tất cả 9 mạng học sâu sau khi được lượng tử hóa sẽ cài đặt trên các MCU bằng ngôn ngữ C dựa trên nền tảng *Tensorflow Lite* trên MCU và được biên dịch bằng trình biên dịch ngôn ngữ C mã nguồn mở tích hợp trên Arduino IDE. Tất cả các ảnh phổ với kích thước khác nhau sẽ được chuyển đổi thành mảng hai chiều kiểu 8 bit cho phù hợp với mạng học sâu lượng tử hóa và lần lượt được nhận dạng. Thời gian thực thi và độ chính xác của tất cả các trường hợp sẽ được ghi nhận lại để phân tích kết quả. Tổng cộng có 1600 ảnh phổ sẽ được kiểm tra trên ba MCU đã chọn.

JTE HCMUTE
ISSN: 1859-1272
JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.hcmute.edu.vn/index.php/jte/index
Email: jte@hcmute.edu.vn

Kết quả thực nghiệm cho thấy các MCU không đủ bộ nhớ để thực thi kiến trúc mạng 1 với tất cả các kích thước ảnh ngõ vào. Đối với mạng học sâu sử dụng kiến trúc 2, các MCU chỉ có thể thực thi mạng ở hai kích thước ảnh 48x48 và 64x64 pixel. Riêng mạng sử dụng kiến trúc 3 thì có thể thực thi với cả ba cỡ ảnh. Kết quả kiểm tra tốc độ và độ chính xác khi thực thi các mạng học sâu sử dụng kiến trúc 2 và 3 trên các MCU được tóm tắt trong *Hình 5*.

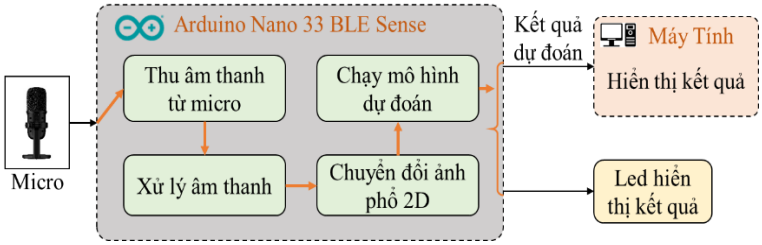


Hình 5. *Độ chính xác và thời gian nhận dạng trên 3 dòng vi điều khiển của: a) các mô hình sử dụng kiến trúc 3 với ba kích cỡ ảnh, b) các mô hình sử dụng kiến trúc 2 với ba kích cỡ ảnh*

Kết quả này chứng minh được mạng học sâu áp dụng kiến trúc 2 với ngõ vào ảnh 64x64 pixel và kiến trúc 3 với ngõ vào 64x64 và 96x96 pixel cho độ chính xác nhận dạng rất cao, đạt trên 98% ngoại trừ trường hợp vi điều khiển nRF52840 thực thi kiến trúc 3 với kích thước ngõ vào 96x96 chỉ đạt 95,5%. Như vậy, mạng học sâu dựa trên kiến trúc 3 với kích thước ngõ vào 64x64 cho kết quả tối ưu nhất vì không chỉ có tổng thời gian thực thi thấp từ 0,33 đến 0,39 giây mà còn có độ chính xác khá ấn tượng từ 97,75% đến 99%. Do đó có thể kết luận đây là kiến trúc mạng phù hợp nhất để thực thi thời gian thực trên các dòng MCU đã chọn. Bên cạnh đó, mặc dù có thời gian thực thi mạng dài nhất, nRF52840 lại cho độ chính xác nổi bậc nhất 99%. Hơn nữa, kit Arduino Nano 33 BLE Sense, hỗ trợ MCU này, cũng được tích hợp sẵn micro rất thuận cho việc thu âm thanh. Chính vì thế, nRF52840 sẽ được chọn đánh giá khả năng chạy thời gian thực của mạng học sâu thay vì là hai MCU còn lại.

3.3 Nhận dạng lỗi động cơ thời gian thực

Hình 6 trình bày sơ đồ khối hệ thống nhận dạng thời gian thực dựa trên kit Arduino Nano 33 BLE Sense. Âm thanh sẽ được thu trực tiếp từ micro tích hợp, nRF52840 thực thi bốn tác vụ chính bao gồm thu một đoạn âm thành có thời lượng 1 giây với tần số lấy mẫu là 16kHz, sau đó sẽ xử lý lọc nhiễu bằng một trong hai bộ lọc Wavelet hoặc MAF. Âm thanh sau lọc sẽ được chuẩn hóa biên độ trước khi chạy thuật toán tạo ảnh phổ 2D như đã minh họa trong Hình 2. Cuối cùng mạng học sâu sẽ được gọi để nhận dạng ảnh phổ vừa tạo và cập nhật xác suất dự đoán ở ngõ ra.

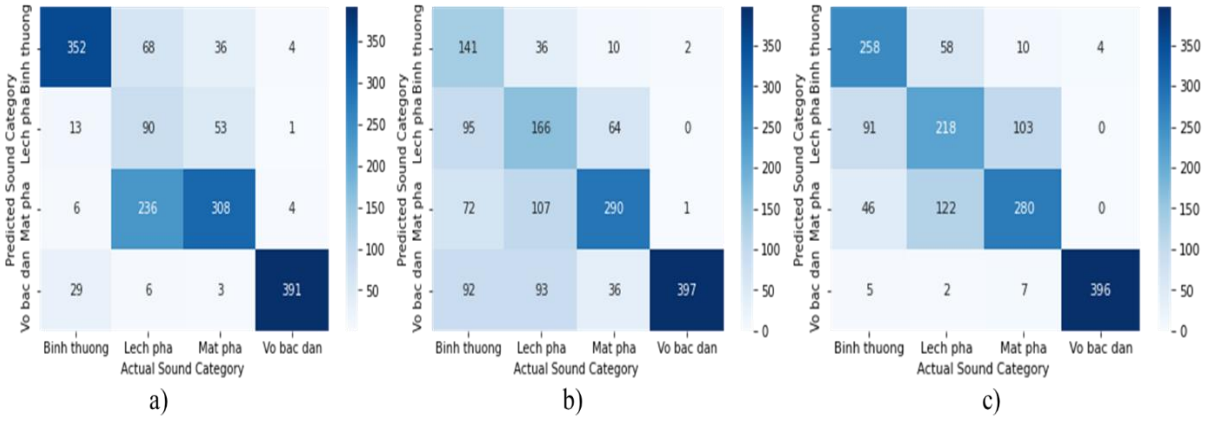


Hình 6. *Sơ đồ mô hình hệ thống nhận diện thời gian thực*

JTE HCMUTE
ISSN: 1859-1272
JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.hcmute.edu.vn/index.php/jte/index
Email: jte@hcmute.edu.vn

Do điều kiện thực tế, nghiên cứu này chưa thực hiện trực tiếp với các âm thanh được phát ra từ động cơ thật. Giải pháp hiện tại của nhóm nghiên cứu là phát các đoạn âm thanh bằng loa và hệ thống nhúng sẽ thu và nhận dạng chúng. Loa phát cần được lựa chọn sao cho âm thanh tái tạo gần với âm thanh thực tế nhất.

Thí nghiệm nhận dạng thời gian thực sẽ được thực hiện lần lượt với từng loại âm thanh. Mỗi loại âm thanh sẽ được MCU nhận dạng 400 lần. Thí nghiệm được thực hiện trong ba trường hợp bao gồm không lọc âm thanh sau khi thu, lọc bằng bộ lọc Wavelet và MAF. Bộ lọc Wavelet sử dụng 4 hệ số Wavelet, ngưỡng mềm $T = 1$; MAF sử dụng cửa sổ trược 16 phần tử. Tất cả các dữ liệu thí nghiệm sẽ được phân tích để tính toán độ chính xác, *F1-score* và vẽ *confusion matrix* để so sánh kết quả của các trường hợp thí nghiệm.



Hình 7. *Kết quả thực nghiệm nhận dạng âm thanh động cơ phát từ loa: a) không lọc tín hiệu âm thanh, b) lọc wavelet và c) lọc trung bình*

Hình 7 trình bày *confusion matrix* của ba trường hợp thí nghiệm. Kết quả cho thấy hiện tượng vỡ bạc đạn đã được mạng học sâu nhận dạng tốt trong tất cả các trường hợp. Âm thanh của hai hiện tượng mất pha và lệch pha dễ bị nhầm lẫn nhau trong cả 3 trường hợp nhưng lọc MAF cho kết quả tốt nhất.

Khi âm thanh không qua bộ lọc (*Hình 7a*) thì âm thanh khi động cơ hoạt động bình thường được nhận dạng rất tốt và nó cũng rất dễ bị nhầm lẫn bởi âm thanh lệch pha (68/400 lần) và mất pha (36/400 lần). Âm thanh khi động cơ mất pha và lệch pha rất dễ bị nhầm lẫn qua lại, đặc biệt là lệch pha bị nhầm thành mất pha 236/400 lần. Khi âm thanh được lọc Wavelet, kết quả nhận dạng có cải thiện nhưng sự nhầm lẫn tăng ở cả ba nhóm âm thanh (*Hình 7b*); thời gian nhận dạng cũng tăng lên do bộ lọc Wavelet khá phức tạp, xử lý chủ yếu trên số thực nên cần nhiều thời gian tính toán.

Bảng 4. *Độ chính xác nhận dạng thời gian thực*

Thuật toán lọc	Độ chính xác tổng quát	F1-score của bình thường	F1-score của lệch pha	F1-score của mất pha	F1-score của vỡ bạc đạn
Không lọc	71,31%	81,86%	32,36%	64,57%	94,33%
Wavelet	62,13%	47,88%	45,79%	66,68%	78%
MAF	**72%**	**70,68%**	**53,64%**	**66,04%**	**97,78%**

Bộ lọc MAF tỏ ra hiệu quả nhất khi không chỉ cải thiện được độ chính xác nhận dạng mà còn giảm được sự nhầm lẫn giữa các loại âm thanh (*Hình 7c*). Bên cạnh đó, tốc độ nhận dạng cũng rất nhanh do bộ lọc này khá đơn giản. Tuy nhiên, sự nhầm lẫn giữa âm thanh lệch pha và mất pha vẫn chưa được cải thiện. Từ *confusion matrix*, độ chính xác tổng quát và *F1-score* của các trường hợp thực nghiệm được minh họa ở *Bảng 4*. Bảng này một lần nữa khẳng định bộ lọc MAF là phù hợp nhất để nhận dạng thời gian thực các loại âm thanh phát ra từ động cơ hay phù hợp nhất để nhận dạng lỗi của động cơ dựa trên tiếng ồn phát ra với độ chính xác tổng quát cao nhất 72%, cải thiện *F1-score* trong cả ba trường hợp lệch pha, mất pha và vỡ bạc đạn so với trường hợp lọc Wavelet và không lọc.

3.4 Thảo luận

Nghiên cứu này đã đề xuất được một kiến trúc mạng học sâu có thể nhận dạng thời gian thực một số lỗi thông dụng trên động cơ điện ba pha. Mặc dù dựa trên kiến trúc mạng CNN (Convolutional Neural Network), kiến trúc đề nghị cho kết quả nhận khá ấn tượng khi thực thi trên máy tính. Độ chính xác của mạng có kiến trúc và ngõ vào lớn nhất trong nghiên cứu này (99,7%) thậm chí cao hơn mạng GoogleNet được áp dụng trong nghiên cứu [3] (98,8%). Nghiên cứu [3] cũng đang bỏ ngỏ khả năng ứng dụng trên hệ thống nhúng và thiết bị di động.

Hạn chế của nghiên cứu này là chỉ dừng lại ở thu thập dữ liệu và thử nghiệm kết quả với một động cơ và chưa đánh giá được tác động của nhiễu từ các nguồn âm thanh khác nhau. Do đó, một cách lý tưởng hệ thống chỉ mới có thể áp dụng với các động cơ hoạt động độc lập và cách xa nguồn nhiễu âm thanh. Trong thời gian tới, nhóm nghiên cứu sẽ thu thập dữ liệu và thử nghiệm với nhiều động cơ khác nhau cũng như tập trung tìm giải pháp khắc phục nhiễu tạp âm trong môi trường thực tế để có thể mở rộng khả năng cũng như phạm vi ứng dụng của nghiên cứu này.

4. Kết luận

Nghiên cứu này đã đề xuất một thủ tục để thiết kế và thực hiện một mạng học sâu trên hệ thống nhúng để nhận dạng lỗi động cơ điện ba pha dựa trên tiếng ồn mà nó tạo ra khi hoạt động. Kết quả thực ngiệm đã tìm ra được hai kiến trúc mạng có thể hoạt động tốt với cả 3 dòng MCU được đề xuất. Kích thước ảnh phổ 64x64 pixel là phù hợp nhất cho kiến trúc mạng học sâu đã đề nghị, cho thời gian thực nhanh nhất là 0,33 giây và độ chính xác cao nhất là 99% với ảnh phổ từ tập dữ liệu được tạo từ âm thanh thu thập sẵn trên máy tính. Song song đó, kết quả thử nghiệm thời gian thực cũng cho kết quả khá ấn tượng, trong trường hợp tín hiệu âm thanh được lọc bằng bộ lọc MAF để loại bỏ nhiễu độ chính xác nhận dạng đạt 72%. Ngoài đạt được mục tiêu đặt ra, nghiên cứu này còn đưa ra một bức tranh tổng quan về kiến trúc điển hình của một mạng học sâu có khả năng thực hiện trên một số MCU tầm trung để làm cơ sở thiết kế mạng cũng như lựa chọn MCU cho các ứng dụng tương tự trong tương lai.

Lời cảm ơn

Nhóm tác giả xin gửi lời cảm ơn chân thành đến ThS. Hoàng Văn Tùng, giảng viên trường Cao đẳng nghề Bạc Liêu, đã trực tiếp thực hiện tạo các sự cố trên động cơ điện, thu âm và chia sẻ tập dữ liệu để nhóm có thể hoàn thành được nghiên cứu này.

TÀI LIỆU THAM KHẢO

[1] H. Henao et al., “Trends in fault diagnosis for electrical machines: A review of diagnostic techniques,” *IEEE Industrial Electronics Magazine*, 8 (2), pp. 31-42, 2014, doi: 10.1109/MIE.2013.2287651.

[2] W. Gong et al., “A novel deep learning method for intelligent fault diagnosis of rotating machinery based on improved CNN-SVM and multichannel data fusion”. *Sensors*, 19, 1693, 2019, doi: 10.3390/s19071693.

[3] H. V. Tung, N. V. Khanh, N. C. Ngon, “Proposal of noninvasive failure diagnosis of electrical motor using googlenet,” *The Journal of Technical Education Science*, no. 66, pp. 3-6, Oct. 2021, doi: 10.54644/jte.66.2021.1070.

[4] Mathworks, “Predictive Maintenance Toolbox.” mathworks.com. https://www.mathworks.com/products/predictive-maintenance.html (accessed Jun. 17, 2022).

[5] A. Géron, “The Machine Learning Landscape,” in *The Fundamentals of Machine Learning,” in Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow*, R. Roumeliotis and N. Tache, 2nd ed, CA, USA: O’Reilly Media, 2019, ch. 1, pp. 22-66.

[6] P. R. Partha, “A review on TinyML: State-of-the-art and prospects,” *Journal of King Saud University – Computer and Information Sciences*, vol. 34, no. 1, pp.1595-1623, Nov. 2021, doi: 10.1016/j.jksuci.2021.11.019.

[7] S. I. Ramon, “LPWAN and embedded machine learning as enablers for the next generation of wearable devices,” *Sensors*, vol.21 , no. 1, pp. 4-8, July. 2021, doi: 10.3390/s21155218.

[8] M. D. Prado et al., “Robustifying the Deployment of tinyML Models for Autonomous Mini-Vehicles,” *Sensors*, vol. 21, no. 1, pp. 4-10, Feb. 2021, doi: 10.3390/s21041339.

[9] Google, “Welcome To Colaboratory.” colab.research.google.com. https://colab.research.google.com/notebooks/welcome.ipynb?hl=en (accessed Jun. 17, 2022).

[10] A. A. Jaber and R.Bicker, “Real-Time Wavelet Analysis of a Vibration Signal Based on Arduino-UNO and LabVIEW,” *International Journal of Materials Science and Engineering*, vol. 3, no. 1, pp. 1-5, March. 2015, doi: 10.12720/ijmse.3.1.66-70.

[11] S. W. Smith, " Moving average filters,” in *The scientist and engineer's guide to digital signal processing*, 2nd ed, CA, USA: California Technical Publishing, 1999, pp. 277-284.

[12] V. Giurgiutiu, “Wave propagation SHM with PWAS transducers,” in Structural Health Monitoring with Piezoelectric Wafer Active Sensors, V. Giurgiutiu, 2nd ed, Academic Press, 2014, pp. 639-706.

[13] Borgerding, “mborgerding/kissfft.” github.com. https://github.com/mborgerding/kissfft (accessed Jun. 13, 2022)

[14] P. Warden and D. Situnayake “Wake-Word Detection: Training a Model,” *in TinyML Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*, 1st ed, CA, USA: O’Reilly Media, 2019, ch. 8, pp. 182-219.

[15] T. Wang et al., "Automatic ECG Classification Using Continuous Wavelet Transform and Convolutional Neural Network," Entropy, vol. 23, pp. 4-12, Jan. 2021, 10.3390/e23010119.
[16] A. Faysal, W. K. Ngui, M. H. Lim, M. H. Lim and M. S. Leong ,"Noise Eliminated Ensemble Empirical Mode Decomposition Scalogram Analysis for Rotating Machinery Fault Diagnosis," Sensors, vol. 21, no. 1, pp. 6-18, Dec. 2021, doi: 10.3390/s21238114.
[17] R. Simon, M. D. Radmacher, K. Dobbin and L. M. McShane, "Pitfalls in the Use of DNA Microarray Data for Diagnostic and Prognostic Classification," *Journal of the National Cancer Institute*, vol. 95, no. 1, pp. 14–18, Jan. 2003, doi: 10.1093/jnci/95.1.14.
[18] S. Abrahams, A. Scarpinelli, D. Hafner, E. Erwitt, "TensorFlow Fundamentals." *in TensorFlow for Machine Intelligence*, Bleeding Edge Press, Santa Rosa, CA 95404, ch. 3, pp. 60-116.
[19] "TensorFlow Lite," tensorflow.org. https://www.tensorflow.org/lite/ (accessed Jun. 17, 2022).
[20] P. E. Novac, G. B. Hacene, A. Pegatoquet, B. Miramond and V. Gripon, "Quantization and Deployment of Deep Neural Networks on Microcontrollers," *Sensors*, vol.21, no1, pp. 5-6, April 2021, doi: 10.3390/s21092984.
[21] "Post-training quantization," tensorflow.org. https://www.tensorflow.org/lite/performance/post_training_quantization (accessed Jun. 17, 2022).

Nguyen Van Khanh received his master degree from Ho Chi Minh University of Technology, Vietnam in 2014 and Doctor of Engineering degree from Tokyo University of Marine Science and Technology, Japan in 2020.
Since 2007, he has been a lecturer at Department of Automation Technology, College of Engineering Technology, Can Tho University. His research interests concentrate on embedded systems, AIoT- and IoT-based applications in environmental and agricultural control.

Tran Vy Khang is a B.S. degree student in Automation and Control Engineering of the Department of Automation Technology, College of Engineering, Can Tho University, Vietnam. He will graduate his B.S. degree at the end of December 2022. Email tranvykhang1906@gmail.com, Contact phone 0706950015.

Nguyen Minh Khai is a B.S. degree student in Automation and Control Engineering of the Department of Automation Technology, College of Engineering, Can Tho University, Vietnam. He will graduate his B.S. degree at the end of December 2022. Email nguyenminhkhai.070500@gmail.com, Contact phone 03698463651.

Thach Van To Em is a B.S. degree student in Automation and Control Engineering of the Department of Automation Technology, College of Engineering, Can Tho University, Vietnam. He will graduate his B.S. degree at the end of September 2022. Email toem2704@gmail.com, Contact phone 0868080442

Pham Tran Lam Hai received his master degree from University of South Australia (UniSA) in 2010.
Since 2012, he has been a lecturer at Department of Automation Technology, College of Engineering Technology, Can Tho University. His research interests focus on Bistatic LIDAR system for gas measurement in environmental and agricultural applications.

Chi-Ngon Nguyen received B.S. and M.S. degrees in Electronic Engineering from Can Tho Universityand the National University, Ho Chi Minh City University of Technology, Vietnam, in 1996 and 2001, respectively. The degree of Ph.D. in Control Engineering was awarded by the University of Rostock, Germany, in 2007.
Since 1996, he has worked at the Can Tho University. He is an associate professor in automation at Department of Automation Technology, and former dean of the College of Engineering at the Can Tho University. Currently, he is a Vice Chairman of the Board of Trustee of Can Tho University.
His research interests are intelligent control, medical control, pattern recognition, classifications, speech recognition, computer vision and agricultural automation