

ĐÁNH GIÁ HIỆU NĂNG CỦA MỘT SỐ CẤU TRÚC MẠNG NEURON SÂU TRONG MÔ HÌNH ĐIỆN TOÁN BIÊN

Nguyễn Đức Minh, Bùi Thị Dân
Học Viện Công Nghệ Bưu Chính Viễn Thông

Tóm tắt— Ngày nay, trí tuệ nhân tạo (AI) và học sâu (DL) đã trở thành một phần không thể thiếu của các công nghệ hiện đại. Tuy nhiên, hầu hết các mô hình học sâu đều đòi hỏi tài nguyên tính toán lớn, khiến chúng không khả thi để triển khai trên các thiết bị có tài nguyên hạn chế, đặc biệt là trên các thiết bị biên mạng. Do đó, nhiều cải tiến đối với kiến trúc học sâu đã được đề xuất để cho phép chúng hoạt động với các giới hạn về tính toán và bộ nhớ. Bài báo này thực hiện một số khảo sát nhằm đánh giá mức độ hoạt động của các mô hình học sâu khác nhau khi được triển khai trên thiết bị biên Raspberry Pi 4, một thiết bị thường có hạn chế về sức mạnh tính toán, bộ nhớ và năng lượng. Đánh giá sẽ tập trung vào các yếu tố quan trọng như độ chính xác, sử dụng bộ nhớ, tốc độ xử lý và độ trễ suy luận; nhằm xác định các mô hình có thể đạt được hiệu suất tối ưu trên các thiết bị biên như Raspberry Pi 4 mà không làm giảm đáng kể độ chính xác hay hiệu suất. Các kết quả khảo sát thu được sẽ được sử dụng để hỗ trợ việc triển khai các mô hình AI trong các môi trường hạn chế về tài nguyên cho các ứng dụng điện toán biên thực tiễn.

Từ khóa— Mạng Neuron sâu, Raspberry Pi 4, ShuffleNet, MNASNet, EfficientNet.

I. GIỚI THIỆU

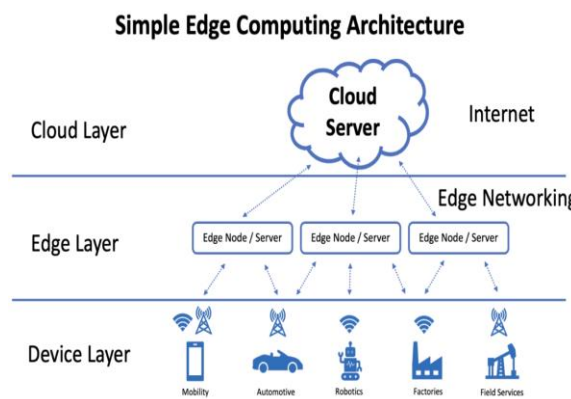
Trong thời đại cách mạng công nghệ 4.0 và sự bùng nổ của IoT thì điện toán biên (Edge Computing) đang được triển khai rộng rãi để đưa ra những ứng dụng, dữ liệu gần hơn với các thiết bị và người dùng nhằm khắc phục những hạn chế của điện toán đám mây (Cloud Computing) về độ trễ, yêu cầu băng thông lớn cũng như khả năng xử lý dữ liệu. Cùng với đó trí tuệ nhân tạo (Artificial Intelligence- AI) và học sâu (Deep Learning-DL) đã trở thành những yếu tố cốt lõi thúc đẩy sự phát triển của các ứng dụng thông minh. Việc áp dụng AI và DL trên các thiết bị biên (Edge Devices) mang lại nhiều lợi ích to lớn, giúp cho chúng trở nên thông minh hơn và giảm tải tính toán cho đám mây. Tuy nhiên, hầu hết các mô hình học sâu khi thực hiện đều đòi hỏi sử dụng tài nguyên phần cứng rất lớn phục vụ cho tính toán, điều này gây khó khăn khi thực thi các mô hình học sâu trên những thiết bị có cấu hình hạn chế như các

thiết bị biên của mạng. Việc phát triển các mô hình học sâu phù hợp với thiết bị biên cũng như tìm kiếm các thiết bị biên có khả năng triển khai những mô hình DL này là một nhu cầu bắt buộc nhằm hiện thực hóa mô hình điện toán biên.

Máy tính nhúng Raspberry Pi 4 là một thiết bị phần cứng phổ biến nhờ kích thước nhỏ gọn và giá thành phải chăng, mang tiềm năng trở thành nền tảng triển khai các mô hình AI trong rất nhiều ứng dụng IoT, tự động hóa và hệ thống nhúng. Tuy nhiên, việc triển khai các mô hình mạng neuron sâu trên Raspberry Pi 4 không chỉ đòi hỏi khả năng tính toán mà còn đòi hỏi tối ưu hóa cấu trúc mạng neuron để đảm bảo hiệu quả về thời gian xử lý, năng lượng tiêu thụ và độ chính xác.

Bài báo này sẽ tập trung vào việc triển khai các mô hình học sâu, so sánh và đánh giá hiệu quả của các cấu trúc mạng neuron sâu khác nhau trên máy tính nhúng Raspberry Pi 4, từ đó đề xuất những giải pháp tối ưu hóa nhằm nâng cao khả năng ứng dụng của AI trên các thiết bị biên của mạng có tài nguyên hạn chế.

A. Thiết bị biên của mạng (Edge devices)



Hình 1. Mô hình kiến trúc điện toán biên

Thiết bị biên của mạng là những thiết bị phần cứng trong mạng IoT được triển khai gần với nguồn dữ liệu hoặc người dùng, thường nằm ở "biên" của mạng, nghĩa là không phải ở trung tâm dữ liệu hay đám mây.

Một số ví dụ về thiết bị biên bao gồm: các cảm biến IoT, camera an ninh, thiết bị điều khiển công nghiệp, hoặc các thiết bị nhỏ gọn như Raspberry Pi.

Tác giả liên hệ: Nguyễn Đức Minh,

Email: minhnd@ptit.edu.vn

Đến tòa soạn: 09/2024, chỉnh sửa: 10/2024, chấp nhận đăng: 10/2024.



Hình 2. Máy tính nhúng Raspberry Pi 4

B. Mô hình mạng neuron sâu

Mạng neuron sâu (Deep Neural Network - DNN) là một loại mô hình học máy (Machine Learning- ML) dựa trên các mạng neuron nhân tạo (Artificial Neural Networks - ANN) có nhiều lớp, thường được sử dụng để giải quyết các bài toán phức tạp liên quan đến nhận dạng, phân loại, và dự đoán.

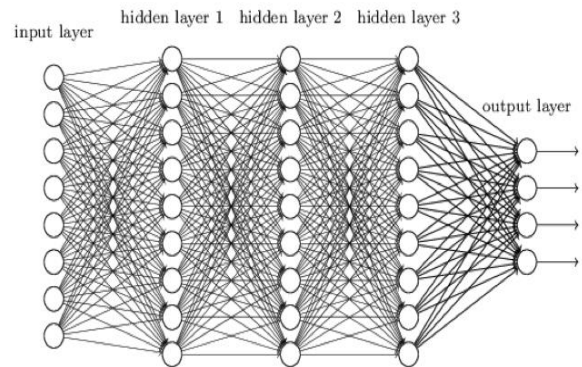
Điểm khác biệt giữa mạng neuron thông thường và mạng neuron sâu là số lượng lớn các lớp ẩn (hidden layers) được sử dụng trong mô hình, giúp nó học được các đặc trưng trừu tượng hơn từ dữ liệu.

Cấu trúc cơ bản của một mạng neuron sâu thường có:

- **Lớp đầu vào (input layer):** Nhận dữ liệu từ bên ngoài, có số lượng neuron tương ứng với số lượng đặc trưng (features) trong dữ liệu.

- **Lớp ẩn (hidden layer):** Gồm nhiều lớp neuron thực hiện phép biến đổi phi tuyến (non-linear transformations) để trích xuất và học các đặc trưng của dữ liệu. Mỗi lớp ẩn được kết nối hoàn toàn với lớp trước đó.

- **Lớp đầu ra (output layer):** xuất kết quả cuối cùng, chẳng hạn như dự đoán phân loại hoặc hồi quy.



Hình 3. Kiến trúc một mạng neuron sâu

Các hàm kích hoạt (activation functions): chẳng hạn như Sigmoid, Tanh hay ReLU [1], được áp dụng sau mỗi lớp để đưa ra các giá trị phi tuyến, giúp mạng có khả năng học những đặc trưng phức tạp.

- **Hàm mất mát (Loss function):** Giúp đánh giá chất lượng của dự đoán và tối ưu mô hình thông qua các thuật toán như Gradient Descent hoặc Adam [2].

Mô hình mạng neuron sâu có rất nhiều ứng dụng trong các lĩnh vực AI, tiêu biểu như nhận dạng hình ảnh (image recognition) trong thị giác máy tính (computer vision), xử lý ngôn ngữ tự nhiên (natural language processing), dự đoán chuỗi thời gian (time series forecasting) ...vv.

II. MÔ HÌNH BÀI TOÁN VÀ PHƯƠNG THỨC THỬ NGHIỆM

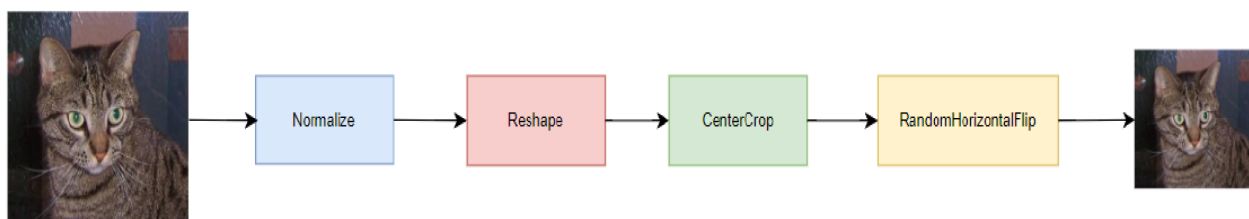
A. Mô hình bài toán

Khi mà các mô hình neuron sâu càng ngày càng trở nên phổ biến, nhu cầu triển khai các mô hình này không chỉ trên đám mây mà còn tại các thiết bị biên của mạng cũng lớn hơn. Tuy nhiên, quá trình này gặp phải nhiều thách thức liên quan đến nguồn tài nguyên tính toán hạn chế của thiết bị biên, bao gồm:

- Hiệu suất xử lý hạn chế: thiết bị biên như Raspberry Pi

Bảng 1: Chú thích và tham số chung cho việc tăng cường dữ liệu

Kỹ thuật	Ý nghĩa	Thông số
Normalize	Chuẩn hóa dữ liệu	Mean = [0.485, 0.456, 0.406] Std = [0.229, 0.224, 0.225]
Reshape	Thay đổi kích thước ảnh	Vuông, 256 × 256
Center crop	Phóng đại vùng trung tâm của ảnh	Vuông, 224 × 224
Random horizontal flip	Lật ngẫu nhiên hình ảnh theo chiều ngang	Tỷ lệ 50%



Hình 4. Các bước tăng cường dữ liệu

Bảng 2: Tham số chung cho huấn luyện mô hình

Tham số/Kỹ thuật	Loại/Thông số
Hàm mất mát (loss function)	Cross-Entropy
Thuật toán tối ưu	Stochastic gradient descent (SGD)
Số vòng lặp (iteration)	30
Tốc độ học (learning rate)	0.1

có CPU và RAM hạn chế, điều này làm cho việc xử lý các mô hình phức tạp trở nên khó khăn và chậm chạp.

- Thiết bị biên thường yêu cầu hoạt động tiết kiệm năng lượng, trong khi các mô hình DNN, đặc biệt là những mô hình lớn, tiêu thụ rất nhiều tài nguyên.

- Khả năng lưu trữ: Mô hình DNN thường yêu cầu dung lượng bộ nhớ lớn, không phù hợp với khả năng lưu trữ hạn chế của thiết bị biên.

Việc triển khai DNN trên các thiết bị biên cũng có rất nhiều lợi ích như: giảm độ trễ so với việc truyền trực tiếp dữ liệu lên đám mây để xử lý, bảo vệ quyền riêng tư của dữ liệu, giảm băng thông, vv. Hiện nay, rất nhiều mô hình DNN đã được thiết kế và tối ưu hóa để có thể sử dụng cho

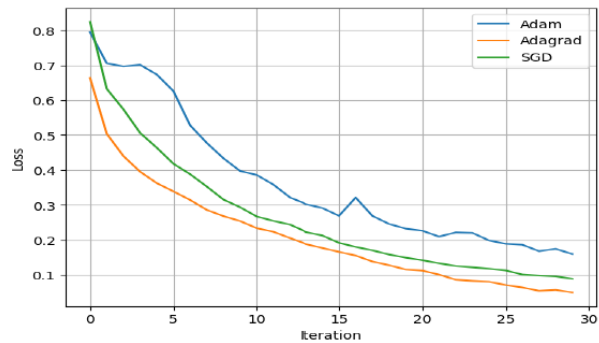
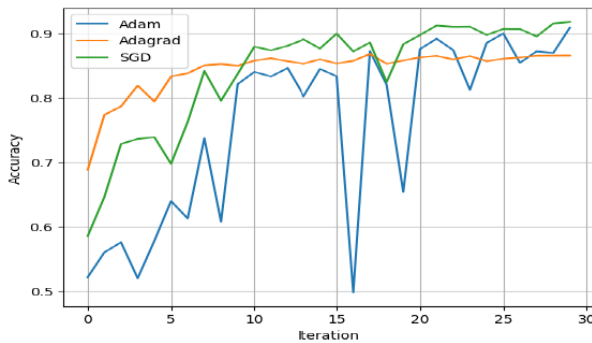
thiết bị nhỏ gọn, hạn chế về khả năng tính toán như MobileNet [3], ShuffleNet [4], EfficientNet [5].

- MobileNet [3] có thể coi là mô hình đầu tiên được giới thiệu trong cuộc cách mạng tối ưu hóa các cấu trúc neuron sâu. Ứng dụng phép toán depthwise separable convolution và kiến trúc inverted residual block, MobileNet đã mở ra cuộc cạnh tranh trong giới nghiên cứu AI về việc tạo ra một mô hình nhỏ, nhẹ nhanh và có hiệu suất tốt.

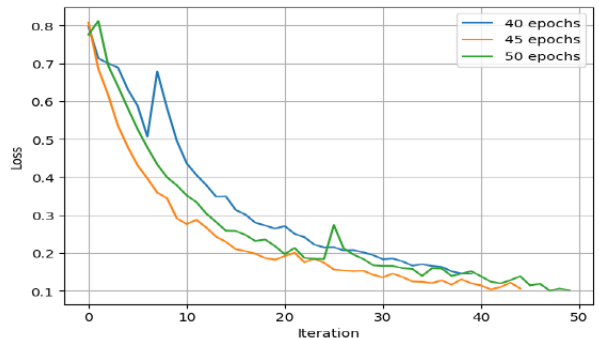
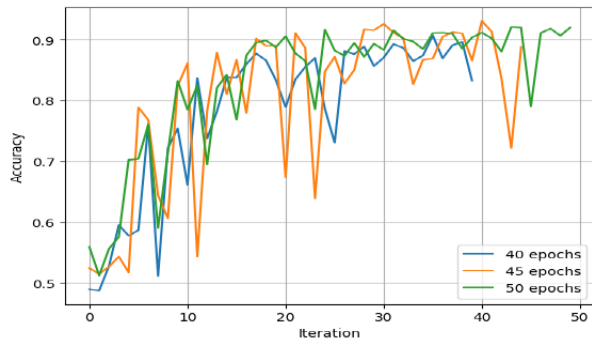
- ShuffleNet [4] được thiết kế với mục tiêu cải tiến MobileNet, giảm hàm lượng tính toán mà vẫn giữ được hiệu suất cao bằng kỹ thuật channel shuffling. Tuy nhiên, độ chính xác của ShuffleNet thường không cao bằng các mô hình lớn hơn nó.

Bảng 3: Thông số đánh giá các mô hình khi sử dụng trên Raspberry Pi 4

Mô hình	Độ chính xác (%)	Số lượng tham số ($\times 10^6$)	MACs ($\times 10^9$)
ShuffleNet-V2	89.36	2.28	0.15
EfficientNet-V2 (nhỏ)	88.53	21.45	2.90
EfficientNet-V2 (lớn)	68.70	118.51	12.38
MobileNet-V3 (nhỏ)	93.17	2.54	0.06
MobileNet-V3 (lớn)	86.91	5.48	0.23
MNASNet 1.0	83.69	4.38	0.33



Hình 4. Thay đổi hàm mất mát và độ chính xác trong quá trình huấn luyện các thuật toán tối ưu khác nhau



Hình 5. Thay đổi hàm mất mát và độ chính xác trong quá trình huấn luyện với số vòng lặp khác nhau

• MNASNet [6] là một mô hình mạng neuron sâu vô cùng độc đáo với khả năng tự động tìm kiếm và thiết kế kiến trúc mạng neuron cho bản thân để tối ưu về hiệu suất và tài nguyên tính toán thông qua kỹ thuật neural architecture search (NAS). MNASNet cũng thừa hưởng các kỹ thuật tối ưu của MobileNet.

• EfficientNet [5] được xem là mô hình mạng neuron sâu mạnh mẽ nhất hiện tại (đạt độ chính xác cao nhất trong rất nhiều tác vụ), được tối ưu bằng phương pháp compound scaling, giúp mở rộng kích thước mô hình theo chiều dài, chiều rộng và độ phân giải một cách cân bằng. Mô hình này có độ chính xác vượt trội, với cái giá là yêu cầu tài nguyên cao hơn.

Do các mô hình trên đều được thiết kế để phù hợp với các thiết bị tính toán nhỏ hơn và hạn chế hơn, việc so sánh độ hiệu quả và yêu cầu tài nguyên giữa chúng khi vận hành trên thiết bị biên là điều cần thiết.

B. Phương thức thử nghiệm

Các mô hình mạng neuron sâu (DNNs) được sử dụng trong bài bao gồm:

- ShuffleNet-V2 [4].
- EfficientNet-V2 [5] (2 phiên bản lớn và nhỏ).
- MobileNet-V3 [3] (2 phiên bản lớn và nhỏ).
- MNASNet 1.0 [6].

Chúng tôi sử dụng phương thức huấn luyện mô hình theo tác vụ phân loại (classification) trên công cụ tính toán độc lập, sau đó trích xuất trọng số mô hình (weights) và sử dụng các trọng số này trên thiết bị biên Raspberry Pi 4. Bộ dữ liệu được sử dụng để huấn luyện và đánh giá các mô hình là bộ dữ liệu DogCat [7]. Tập dữ liệu này bao gồm 25000 ảnh chó và mèo, sẽ được chia làm 2 phần: tập dữ liệu huấn luyện (training) (70% tổng số dữ liệu ảnh, sử dụng để huấn luyện) và tập kiểm tra (test) (30% tổng số dữ liệu ảnh, dùng để kiểm tra). Chúng tôi tiến hành tăng cường dữ liệu (data augmentation) trên tập training để gia tăng độ phong phú của hình ảnh (mô phỏng như Hình 3 và Bảng 1). Các tham số huấn luyện có ở Bảng 2.

Sau đó, chúng tôi tiến hành so sánh và đánh giá độ hiệu quả của các mô hình thông qua các chỉ số sau:

- Tham số mô hình (Parameter): mô hình càng ít tham số thì càng tiết kiệm về không gian nhớ, tuy nhiên sẽ thiệt về độ chính xác so với các mô hình lớn, nhiều tham số hơn.
- Multiply-Accumulation Operation (MAC): một phép tính toán được thực hiện bởi các mô hình trong quá trình suy luận (inference). Mô hình nào càng thực hiện nhiều MACs hơn trong quá trình sẽ tốn nhiều tài nguyên tính toán hơn.
- Độ chính xác (Accuracy): Độ chính xác của các mô hình sẽ được đánh giá trên tập test của bộ dữ liệu DogCat.

III. KẾT QUẢ VÀ MỘT SỐ NHẬN XÉT

Mẫu thiết bị biên mà chúng tôi sử dụng là Raspberry Pi 4 Model B (4GB RAM) với chip Broadcom BCM27111, bộ xử lý Quad-core Cortex-A72 (ARM v8). Kết quả đánh giá các mô hình có ở Bảng 3. Kết quả so sánh các mô hình cho thấy sự khác biệt rõ rệt trong các chỉ số. Về mặt hiệu suất, điều khá bất ngờ là MobileNet-V3 nhỏ lại là mô hình có độ chính xác trên tập test cao nhất (93.17%), trong khi EfficientNet-V2 lại có độ chính xác rất thấp (68.17%).

Điều này cho thấy một thế mạnh của các mô hình có cấu trúc nhỏ gọn: không cần tiêu tốn quá nhiều thời gian huấn luyện để đạt kết quả cao (30 vòng lặp). MobileNet-V3 nhỏ cũng đồng thời là mô hình cần ít phép tính toán nhất với 6×107 MACs. ShuffleNet-V2 nhỏ đứng thứ hai về độ chính xác với 89.36% độ chính xác nhưng chỉ sở hữu 2.28×106 tham số mô hình - một lựa chọn hoàn hảo khác cho các thiết bị biên. Ngược lại, các mô hình lớn hơn vốn cần nhiều thời gian để tối ưu như hai phiên bản EfficientNet-V2 hay MobileNet-V3 lớn đều cho thấy độ hiệu quả kém hơn khi bị hạn chế về tài nguyên.

Thêm vào đó, nhóm tác giả đã khảo sát mô hình MobileNet-V3 nhỏ với các cấu hình khác nhau để tìm ra lựa chọn tối ưu. Đầu tiên, chúng tôi tiến hành thử nghiệm với 3 thuật toán tối ưu khác nhau: Adam [2], Adagrad [8] và SGD [9]. Chúng tôi giữ nguyên tốc độ học bằng 0.01 và số vòng lặp bằng 30. Với mỗi thuật toán, chúng tôi tiến

Bảng 4: Thông số đánh giá các mô hình MobileNet-V3 sau 30 vòng lặp

Thuật toán	Độ chính xác	Hàm mất mát
Adam	90,83%	0,159
Adagrad	86,49%	0,049
SGD	91,71%	0,088

hành 3 lần chạy thử nghiệm và lấy kết quả trung bình. Kết quả thu được như ở Hình 4 và Bảng 4 cho thấy sự hiệu quả của hai thuật toán Adam và SGD (90.83% và 91.71% trên tập test). Ngoài ra, có thể thấy thuật toán Adagrad giúp mô hình đạt đến ngưỡng hội tụ khá sớm và ổn định: ngay từ vòng lặp thứ 5 thì độ chính xác đã không thay đổi nhiều. Hàm mất mát cross-entropy cũng giảm nhanh và đều nhất khi mô hình sử dụng Adagrad. Trong khi đó, cả Adam và SGD đều cho thấy sự dao động khá đáng kể về độ chính xác trong 30 vòng lặp. Như ta đã thấy, thuật toán SGD chưa thể làm mô hình hội tụ ở vòng lặp thứ 30. Để kiểm tra vùng hội tụ của thuật toán SGD, chúng tôi quyết định huấn luyện 3 lần với số vòng lặp khác nhau: 40, 45 và 50. Hàm mất mát cross-entropy được giữ không đổi và tốc độ học đặt bằng 0.01. Từ kết quả ở Hình 5 và Bảng 4, ta có thể thấy rằng dù hàm mất mát của mô hình đã đạt đến ngưỡng hội tụ (không còn giảm nhiều) nhưng độ chính xác vẫn biến thiên rất mạnh. Mô hình của chúng ta có thể đang overfit - mô hình có thể đã học được cả những phân thông tin nhiễu và không quan trọng trong ảnh. Ngoài ra, ta có thể khẳng định rằng SGD không phải là thuật toán tối ưu phù hợp nhất cho MobileNet-V3 do sự thiếu ổn định của nó.

IV. KẾT LUẬN

Trong đề tài này, nhóm tác giả đã thực hiện nghiên cứu và so sánh, đánh giá độ hiệu quả của các cấu trúc mạng neuron sâu (DNNs) trên thiết bị biên Raspberry Pi 4. Kết quả phân tích cho thấy rằng hiệu năng của các mô hình DNN khi thực hiện trên các thiết bị biên của mạng với tài nguyên hạn chế không chỉ phụ thuộc vào sự phức tạp của mô hình DNN và tài nguyên phần cứng hạn chế của Raspberry Pi 4, mà còn phụ thuộc vào cách huấn luyện và khả năng lựa chọn các thuật toán, các tham số tốt nhất.

Nghiên cứu cũng khẳng định rằng việc tối ưu hóa mạng neuron sâu đã đạt được những kết quả đáng ghi nhận:

những mô hình DNN được thiết kế dành cho các thiết bị với tài nguyên hạn chế đều tỏ ra hiệu quả cả về độ chính xác hay tài nguyên tiêu tốn. Những mô hình gọn nhẹ như MobileNet-V3 hay ShuffleNet đều đạt được độ chính xác đáng kể và cao hơn so với các mô hình có độ phức tạp cao hơn chúng như EfficientNet-V2. Điều này là cơ sở vững chắc cho các ứng dụng AI và IoT khi triển khai thực tế trên các thiết bị biên của mạng có tài nguyên hạn chế.

TÀI LIỆU THAM KHẢO

- [1] V. Nair and G. Hilton, "Rectified linear units improve restricted Boltzmann machines", 2010.
- [2] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization", 2017.
- [3] A. Howard, M. Sandler, G.Chu, L.C. Chen, B. Chen, M.Tan, W.Wang, Y.Zhu, R. Pang, V. Vasudevan, Q. L. Le, and H. Adam, "Searching for mobilenet v3", 2019.
- [4] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," 2018.
- [5] M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," 2021.
- [6] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," 2019.
- [7] W. Cukierski, "Dogs vs. cats redux: Kernels edition," 2016.
- [8] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," 2011.
- [9] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016

PERFORMANCE EVALUATION OF SOME DEEP NEURAL NETWORK STRUCTURES IN EDGE COMPUTING MODELS

Abstract— Nowadays, artificial intelligence and deep learning has become an integral part of modern technologies. However, most deep learning models require extensive computation resources, making them infeasible for deployment with limited resources, especially on edge devices. As a result, several innovations to deep learning architectures have been proposed to allow them to operate with constrained computation and memory capabilities. This project aims to assess how well various deep learning models perform when deployed on a well-known edge device, namely Raspberry Pi 4, which typically has constraints in computational power, memory, and energy. The evaluation will focus on key factors like accuracy, memory usage, processing speed and inference latency; aiming to identify the models that can achieve optimal performance on edge devices like the Raspberry Pi 4 without significant trade-offs in accuracy or efficiency. The findings are intended to guide the deployment of AI models in resource-constrained environments for real-world applications.

Keywords—Deep Neuron Network, Rasberry Pi 4, ShuffleNet, MNASNet, EfficientNet.



Lĩnh vực nghiên cứu: Radar nhiều vị trí, xử lý tín hiệu, điện tử máy tính, IoT và học máy.



Lĩnh vực nghiên cứu: Xử lý tín hiệu, điện tử máy tính, IoT và học máy.

Nguyễn Đức Minh tốt nghiệp đại học chuyên ngành Vật lý-Vô tuyến năm 2000 tại Đại học Khoa học tự nhiên- Đại học Quốc gia Hà Nội. Tốt nghiệp Thạc sỹ ngành Công nghệ Điện tử - Viễn thông năm 2006 tại Đại học Công nghệ - Đại học Quốc gia Hà Nội. Nhận bằng tiến sĩ kỹ thuật ngành Kỹ thuật Ra đa-Dẫn đường năm 2019 tại Học viện Kỹ thuật Quân sự. Hiện đang giảng dạy tại Học viện Công nghệ Bưu chính Viễn thông.

Bùi Thị Dân tốt nghiệp đại học chuyên ngành Điện tử- Viễn thông năm 2000 tại Đại học Giao thông Vận tải. Tốt nghiệp Thạc sỹ ngành Công nghệ Điện tử - Viễn thông năm 2007 tại Đại học Công nghệ - Đại học Quốc gia Hà Nội. Hiện đang giảng dạy tại khoa Kỹ thuật Điện tử 1 - Học viện Công nghệ Bưu chính Viễn thông.