Tạp chí Khoa học và Công nghệ Trường Đại học Công nghệ Đông Á

So sánh hiệu năng và chi phí tính toán của một số kiến trúc mạng nơ-ron tích chập nhẹ trên tập dữ liệu CIFAR-10

Comparative Analysis of Performance and Computational Costs of Lightweight Convolutional Neural Network Architectures on CIFAR-10

Trần Nguyên Hoàng

Khoa Công nghệ thông tin, Trường Đại học Công nghệ Đông Á

Email: hoangtn@eaut.edu.vn

TÓM TẮT

Bài báo trình bày một nghiên cứu thực nghiệm có kiểm soát nhằm so sánh mối quan hệ giữa hiệu năng phân loại và chi phí tính toán của năm kiến trúc mạng nơ-ron tích chập trên tập dữ liệu CIFAR-10, gồm SimpleCNN, VGG-11-Tiny-BN, ResNet18-CIFAR, MobileNetV2 và EfficientNet-B0. Các mô hình được huấn luyện trong cùng một thiết lập thống nhất: tiền xử lý, tăng cường dữ liệu, tối ưu bằng SGD với lịch điều chỉnh tốc độ học dạng cosine giảm dần, làm mượt nhãn và đánh giá lặp lại trên năm hạt giống ngẫu nhiên. Hiệu năng được đánh giá bằng độ chính xác và F1-macro trên tập kiểm thử; chi phí mô hình được định lượng bằng số tham số, FLOPs và độ trễ suy luận trên CPU.

Kết quả cho thấy ResNet18-CIFAR đạt hiệu năng phân loại cao nhất, với độ chính xác và F1-macro đều đạt 94,68 ± 0,23%, nhưng đồng thời có số tham số, FLOPs và độ trễ CPU cao nhất. Ngược lại, SimpleCNN đạt độ chính xác 91,16 ± 0,08% và F1-macro 91,15 ± 0,08%, thấp hơn ResNet18-CIFAR nhưng có số tham số nhỏ nhất và độ trễ suy luận thấp nhất. Điều này cho thấy khi lựa chọn mô hình cho bài toán ảnh nhỏ và môi trường tài nguyên hạn chế, cần cân nhắc đồng thời độ chính xác và chi phí triển khai. Bài báo không đề xuất kiến trúc mới mà cung cấp thêm bằng chứng thực nghiệm về sự đánh đổi giữa hiệu năng và tài nguyên của các kiến trúc CNN trong bối cảnh CIFAR-10.

Từ khóa: chi phí tính toán; CIFAR-10; độ trễ suy luận; mạng nơ-ron tích chập; phân loại ảnh

ABSTRACT

This study presents a controlled experimental comparison of five convolutional neural network architectures on the CIFAR-10 benchmark dataset, including SimpleCNN, VGG-11-Tiny-BN, ResNet18-CIFAR, MobileNetV2, and EfficientNet-B0. The objective is to examine the relationship between classification performance and computational cost in small-image classification. All models were trained under the same preprocessing, augmentation, optimization protocol and evaluated over five random seeds. Performance was assessed using test accuracy and macro-averaged F1-score, while computational efficiency was quantified using the number of learnable parameters, floating-point operations and CPU inference latency. The results show that ResNet18-CIFAR achieved the highest classification performance, with 94.68 ± 0.23% accuracy and 94.68 ± 0.23% macro-F1, but it also required the largest number of parameters, FLOPs and inference time. SimpleCNN achieved lower but

DOI: https://doi.org/10.65153/5vswxq05

still competitive accuracy, while having the smallest model size and lowest CPU latency. These findings indicate that model selection should consider both predictive performance and deployment cost rather than accuracy alone.

Keywords: CIFAR-10, convolutional neural networks, computational cost, image classification, inference latency.

1. MỞ ĐẦU

Tập dữ liệu CIFAR-10 [1] là bộ dữ liệu chuẩn cho bài toán phân loại ảnh kích thước nhỏ, vừa đủ đơn giản để huấn luyện nhanh, vừa đủ khó để đánh giá khả năng khái quát hóa của mô hình. Bài báo này không đề xuất một kiến trúc hoặc thuật toán huấn luyện mới, mà tập trung vào so sánh thực nghiệm có kiểm soát giữa năm kiến trúc CNN đại diện. Mục tiêu là đánh giá đồng thời độ chính xác, F1-macro, số tham số, FLOPs và độ trễ suy luận trên CPU, từ đó cung cấp thêm căn cứ lựa chọn mô hình trong bối cảnh ảnh nhỏ, giảng dạy, phòng thí nghiệm và hệ thống tài nguyên hạn chế.

Trong những năm gần đây, học sâu và đặc biệt là mạng nơ-ron tích chập (CNN) đã trở thành nền tảng cho nhiều hệ thống thị giác máy tính nhờ khả năng trích chọn đặc trưng mạnh và hiệu quả tính toán [2], [3], [4], [5], [6], [7]. Tuy nhiên, phần lớn các nghiên cứu tập trung vào các mô hình rất sâu, dung lượng lớn trên các bộ dữ liệu quy mô cao, trong khi nhu cầu triển khai mô hình gọn nhẹ cho ứng dụng nhỏ, hệ thống nhúng và môi trường phòng thí nghiệm vẫn ngày càng tăng.

2. PHƯƠNG PHÁP VÀ THIẾT KẾ THÍ NGHIỆM

2.1. Công trình liên quan

Trong hơn một thập kỷ qua, sự phát triển của các kiến trúc mạng nơ-ron tích chập (CNN) đã tạo nên bước tiến quan trọng trong thị giác máy tính, từ các kiến trúc sâu tham số lớn như VGG [2] và ResNet [3] đến các mô hình tối ưu cho thiết bị di động như MobileNetV2 [4] và EfficientNet [5]. VGG sử dụng chuỗi khối tích chập 3×3 đơn giản nhưng số tham số rất lớn, trong khi ResNet đưa vào khối dư (residual block) giúp huấn luyện mạng sâu ổn định hơn. Ngược lại, MobileNetV2 khai thác tích chập phân tách theo chiều sâu (depthwise separable convolution) và khối đảo ngược (inverted residual) để giảm mạnh FLOPs, còn EfficientNet-B0 sử dụng MBConv kết hợp compound scaling để đạt độ chính xác cạnh tranh với chi phí tính toán thấp hơn [4], [5]. Nhờ đó, mỗi họ kiến trúc đại diện cho một triết lý thiết kế khác nhau trên trục từ mô hình "nặng nhưng mạnh" đến mô hình nhẹ cho tài nguyên hạn chế.

Song song với kiến trúc, nhiều công trình tập trung hoàn thiện các thực hành huấn luyện cho CNN như tăng cường dữ liệu và chính quy hóa, ví dụ mixup và AutoAugment [8], [9], tối ưu hóa với AdamW [10] cùng các lịch điều chỉnh tốc độ học hiện đại như cosine có khởi động

lại [11]. Phần lớn các nghiên cứu này được đánh giá trên bộ dữ liệu quy mô rất lớn như ImageNet, trong khi nhiều bối cảnh ứng dụng thực tế lại cần mô hình gọn nhẹ hoạt động hiệu quả trên các tập dữ liệu vừa và nhỏ như CIFAR-10 [1]. Nghiên cứu này kế thừa các hướng tiếp cận nói trên nhưng tập trung vào việc so sánh có hệ thống một nhóm kiến trúc CNN đại diện trong điều kiện tài nguyên hạn chế, qua đó làm rõ sự đánh đổi giữa hiệu năng và tài nguyên khi lựa chọn mô hình cho bài toán phân loại ảnh.

2.2. Dữ liệu và tiền xử lý

Trong nghiên cứu này, chúng tôi sử dụng bộ dữ liệu CIFAR-10 [1] gồm 60 000 ảnh màu kích thước 32×32, trong đó 50 000 ảnh cho tập huấn luyện và 10 000 ảnh cho tập kiểm thử, với 10 lớp đối tượng thường gặp như máy bay, ô tô, chim, mèo, chó, ngựa, tàu thủy và xe tải. Để đánh giá khách quan hơn, 10% dữ liệu của tập huấn luyện được tách ra làm tập kiểm định với hạt giống ngẫu nhiên cố định.

Các ảnh CIFAR-10 được tiền xử lý theo các thực hành huấn luyện mạng nơ-ron tích chập hiện đại [6], [9]. Cụ thể, trên tập huấn luyện, mỗi ảnh được đệm thêm 4 điểm ảnh theo mỗi chiều, sau đó cắt ngẫu nhiên về 32×32 và lật ngang ngẫu nhiên với xác suất 0,5; cuối cùng chuẩn hóa theo giá trị trung bình và độ lệch chuẩn của CIFAR-10 (mean = (0,4914; 0,4822; 0,4465), std = (0,2470; 0,2435; 0,2616)). Trên tập kiểm định và tập kiểm thử, chỉ áp dụng chuẩn hóa, không áp dụng tăng cường dữ liệu. Quy trình này giúp mô hình bền vững hơn trước các biến thiên nhỏ về dịch chuyển và cải thiện khả năng khái quát hóa trên các tập ảnh kích thước nhỏ như CIFAR-10.

2.3. Lựa chọn mô hình

Trong nghiên cứu này, chúng tôi lựa chọn và so sánh năm kiến trúc mạng nơ-ron tích chập tiêu biểu trên tập dữ liệu CIFAR-10 [1]: SimpleCNN, VGG-11-Tiny-BN, ResNet18-CIFAR, MobileNetV2 và EfficientNet-B0. SimpleCNN đóng vai trò mô hình cơ sở để so sánh với các kiến trúc sâu hoặc tối ưu hóa hiện đại hơn. Các mô hình còn lại đại diện cho các hướng thiết kế khác nhau: khối tích chập xếp chồng, kết nối dư, tích chập phân tách theo chiều sâu và chiến lược mở rộng mô hình.

Để bảo đảm khả năng tái lập, cấu trúc của từng mô hình được mô tả theo thứ tự các khối xử lý trong Bảng 1. Với các mô hình chuẩn, lớp phân loại cuối được thay bằng lớp tuyến tính 10 đầu ra. ResNet18-CIFAR đồng thời dùng stem phù hợp ảnh 32×32 (tích chập 3×3, stride 1 và bỏ max-pooling đầu mạng). SimpleCNN và VGG-11-Tiny-BN là các biến thể gọn được cài đặt riêng; ký hiệu Conv(k×k, c) chỉ lớp tích chập kích thước k×k với c kênh đầu ra, BN là chuẩn hóa theo lô, MP là max-pooling, GAP là global average pooling và FC là lớp kết nối đầy đủ.

Bảng 1. **Cấu trúc các mô hình được sử dụng trong thí nghiệm**

Mô hình	Cấu trúc theo thứ tự xử lý	Điều chỉnh cho CIFAR-10
SimpleCNN	3×(Conv 3×3–ReLU–MP) → flatten → FC → FC(10).	Đầu vào RGB 32×32; 5 lớp có tham số học.
VGG-11-Tiny-BN	8 Conv 3×3 trong 5 khối [1–1–2–2–2]; Conv–BN–ReLU; MP cuối khối; bộ phân loại 10 lớp.	Giữ mẫu VGG-11, thu hẹp kênh và bộ phân loại.
ResNet18-CIFAR	Stem Conv 3×3; bốn stage BasicBlock có số khối [2,2,2,2] và số kênh [64,128,256,512]; GAP; FC(10).	Stride stem = 1 và bỏ max-pooling đầu mạng; downsampling ở đầu các stage 2–4.
MobileNetV2	Stem Conv 3×3; inverted residual (t,c,n,s): (1,16,1,1), (6,24,2,2), (6,32,3,2), (6,64,4,2), (6,96,3,1), (6,160,3,2), (6,320,1,1); Conv 1×1; GAP; FC(10).	Width multiplier 1,0; bộ phân loại 10 đầu ra.
EfficientNet-B0	Stem Conv 3×3; bảy stage MBConv1/MBConv6 có (k,c,n,s): (3,16,1,1), (3,24,2,2), (5,40,2,2), (3,80,3,2), (5,112,3,1), (5,192,4,2), (3,320,1,1); head Conv 1×1; GAP; dropout; FC(10).	Giữ cấu hình B0; thay lớp phân loại thành 10 đầu ra.

Ghi chú: t là hệ số mở rộng kênh, c là số kênh đầu ra, n là số lần lặp khối và s là stride của khối đầu trong mỗi stage.

Cần lưu ý rằng MobileNetV2 và EfficientNet-B0 vốn được thiết kế chủ yếu cho ảnh có độ phân giải lớn hơn, trong khi CIFAR-10 chỉ gồm ảnh 32×32. Vì vậy, kết quả trong bài báo được diễn giải trong phạm vi thiết lập thí nghiệm cụ thể, không xem là kết luận tuyệt đối về năng lực tổng quát của từng kiến trúc. Việc giữ cùng quy trình huấn luyện nhằm tạo điều kiện so sánh có kiểm soát, nhưng chưa phải là cấu hình tối ưu riêng cho từng mô hình.

2.4. Thiết lập huấn luyện

Bảng 2. **Cấu hình mô hình và siêu tham số huấn luyện**

Mô hình	Số lớp	Kích thước lô	Số vòng huấn luyện	Bộ tối ưu	Lịch học
SimpleCNN (mô hình cơ sở)	5	128	100	SGD	Cosine giảm dần
VGG-11-Tiny-BN	11	128	100	SGD	Cosine giảm dần
ResNet18-CIFAR	18	128	100	SGD	Cosine giảm dần
MobileNetV2	53	128	100	SGD	Cosine giảm dần
EfficientNet-B0	82	128	100	SGD	Cosine giảm dần

Bảng 2 tóm tắt cấu hình huấn luyện được áp dụng thống nhất cho cả năm mô hình CNN trên tập dữ liệu CIFAR-10, gồm kích thước lô = 128, 100 vòng huấn luyện, bộ tối ưu SGD với tốc độ học khởi tạo 0,1, hệ số quán tính 0,9, suy giảm trọng số 5×10^{-4}, làm mượt nhãn = 0,1 và

DOI: https://doi.org/10.65153/5vswxq05

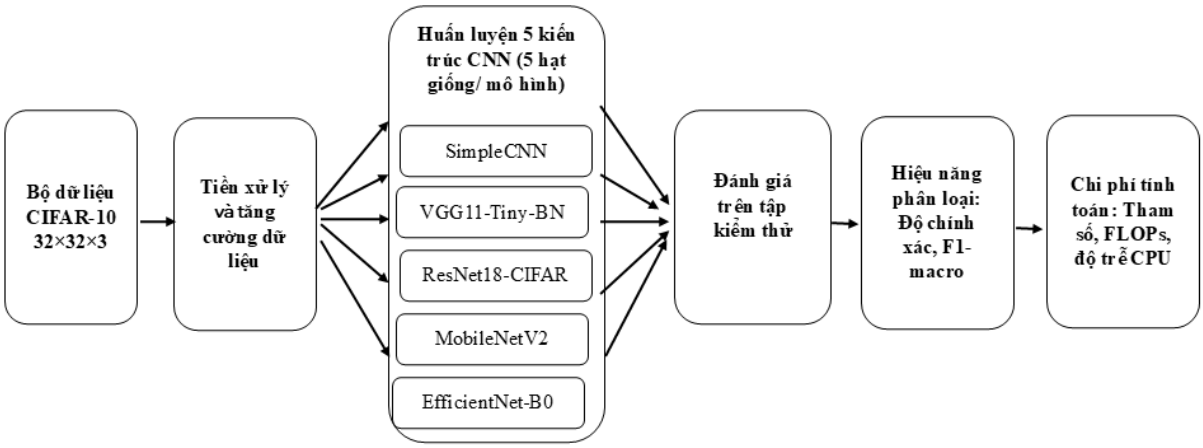
lịch điều chỉnh tốc độ học dạng cosine giảm dần. Mỗi mô hình được huấn luyện lặp lại với năm hạt giống ngẫu nhiên: 42, 43, 44, 45 và 46. Ký hiệu ± trong các bảng kết quả biểu thị độ lệch chuẩn qua năm lần chạy.

Thí nghiệm được thực hiện trên Google Colab với Python 3.12.13, PyTorch 2.11.0+cu128, torchvision 0.26.0+cu128, CUDA khả dụng, GPU Tesla T4, CPU Intel Xeon 2 vCPU @ 2.00 GHz và RAM 12 GB. Độ trễ suy luận được đo trên CPU theo đơn vị ms/ảnh trong cùng điều kiện phần mềm và kích thước lô đánh giá. Cách đo này giúp so sánh tương đối giữa các mô hình trong một môi trường thống nhất, nhưng chưa đại diện cho mọi nền tảng triển khai thực tế.

2.5. Thiết kế thí nghiệm

Các thí nghiệm được thiết kế nhằm đánh giá toàn diện hiệu năng của năm kiến trúc CNN trên tập dữ liệu CIFAR-10 [1] theo hai nhóm tiêu chí chính: (i) chất lượng phân loại thông qua độ chính xác và F1-macro trên tập kiểm thử; (ii) chi phí tính toán thông qua số tham số, số phép toán dấu chấm động và độ trễ suy luận trên CPU. Toàn bộ mô hình được huấn luyện theo cùng quy trình để giảm nhiễu do khác biệt cài đặt; tuy nhiên, bài báo không khẳng định cấu hình này là tối ưu riêng cho từng kiến trúc.

Với mỗi kiến trúc, chúng tôi báo cáo giá trị trung bình và độ lệch chuẩn của độ chính xác và F1-macro trên năm lần chạy. Bảng 3 tóm tắt độ phức tạp mô hình và độ trễ suy luận, còn Bảng 4 trình bày hiệu năng phân loại. Các hình minh họa bổ sung gồm quy trình thí nghiệm, biểu đồ độ chính xác/F1-macro, ma trận nhầm lẫn của mô hình có hiệu năng cao nhất và quan hệ giữa độ chính xác với độ trễ suy luận. Đường cong huấn luyện và kiểm định được lập trực tiếp từ lịch sử 100 epoch của seed 42, được chọn làm lần chạy đại diện và không dùng để thay thế các thống kê trung bình qua năm seed ở Bảng 4.



Hình 1. **Quy trình thí nghiệm so sánh các mô hình CNN trên CIFAR-10.**

Quy trình thí nghiệm bắt đầu từ bộ dữ liệu CIFAR-10, sau đó thực hiện tiền xử lý và tăng cường dữ liệu trước khi huấn luyện năm kiến trúc CNN với năm hạt giống ngẫu nhiên cho mỗi mô hình. Các mô hình sau huấn luyện được đánh giá trên tập kiểm thử theo hai nhóm tiêu chí: hiệu năng phân loại, gồm độ chính xác và F1-macro; và chi phí tính toán, gồm số tham số, FLOPs và độ trễ suy luận trên CPU.

Bảng 3. **Tham số và độ phức tạp tính toán của các mô hình**

Mô hình	Số tham số (triệu)	FLOPs (GMac)	Độ trễ CPU (ms/ảnh)
SimpleCNN	0,551	0,039	3,86 ± 1,01
VGG-11-Tiny-BN	2,310	0,039	5,42 ± 1,19
ResNet18-CIFAR	11,174	0,558	24,96 ± 3,39
MobileNetV2	2,237	0,027	8,96 ± 1,81
EfficientNet-B0	4,020	0,034	14,72 ± 2,80

Bảng 3 cho thấy không thể gọi một mô hình là "tốt nhất về chi phí" nếu không nêu rõ thước đo. SimpleCNN có số tham số nhỏ nhất (0,551 triệu) và độ trễ CPU thấp nhất (3,86 ± 1,01 ms/ảnh), nhưng không có MACs thấp nhất. MobileNetV2 đạt 0,027 GMac, thấp hơn 30,8% so với 0,039 GMac của SimpleCNN; đổi lại, MobileNetV2 có 2,237 triệu tham số (gấp khoảng 4,06 lần) và độ trễ 8,96 ± 1,81 ms/ảnh (gấp khoảng 2,32 lần). Vì vậy, kết luận chính xác định SimpleCNN có lợi thế về dung lượng và độ trễ CPU, còn MobileNetV2 có lợi thế về MACs lý thuyết. Sự khác biệt giữa MACs và độ trễ thực tế có thể xuất phát từ kiểu phép toán, truy cập bộ nhớ và mức tối ưu của các khối depthwise/inverted residual trên CPU và ảnh 32×32.

3. KẾT QUẢ VÀ THẢO LUẬN

3.1. Kết quả tổng quan

Kết quả 5 lần chạy cho thấy ResNet18-CIFAR đạt hiệu năng phân loại cao nhất trên CIFAR-10, với độ chính xác và F1-macro đều đạt 94,68 ± 0,23%. SimpleCNN đạt độ chính xác 91,16 ± 0,08% và F1-macro 91,15 ± 0,08%; tuy không cao nhất về độ chính xác, mô hình này có số tham số và độ trễ thấp nhất. Do đó, kết luận chính của nghiên cứu được điều chỉnh theo hướng: ResNet18-CIFAR tốt nhất về hiệu năng phân loại, còn SimpleCNN có số tham số và độ trễ thấp nhất; MobileNetV2 có MACs thấp nhất trong thiết lập thí nghiệm này.

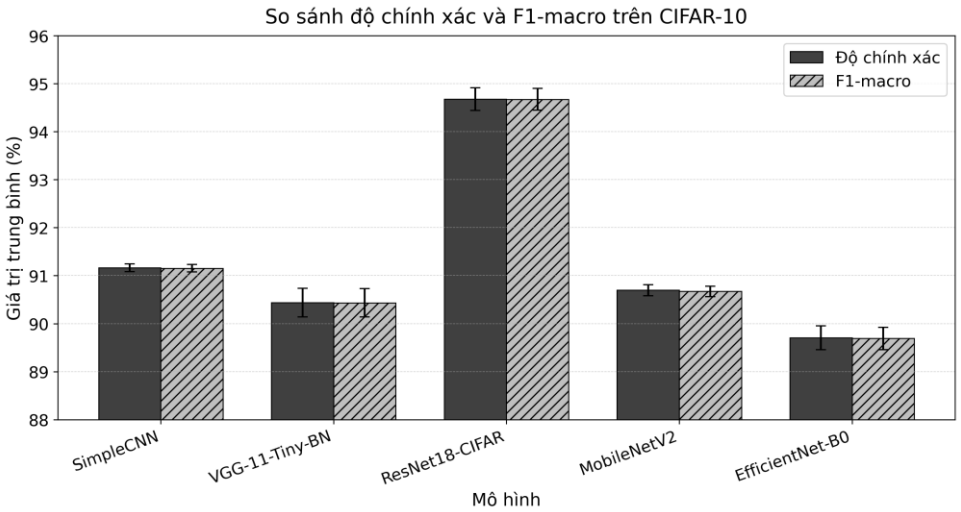
VGG-11-Tiny-BN và MobileNetV2 đạt mức hiệu năng khá gần SimpleCNN, lần lượt đạt 90,44 ± 0,30% và 90,70 ± 0,12% về độ chính xác. Tuy nhiên, VGG-11-Tiny-BN có số tham số cao hơn SimpleCNN, còn MobileNetV2 dù có FLOPs thấp lại không đạt độ trễ thấp nhất trên CPU. EfficientNet-B0 đạt 89,71 ± 0,25% về độ chính xác, thấp hơn các mô hình còn lại trong thiết lập này.

DOI: https://doi.org/10.65153/5vswxq05

Kết quả này không hàm ý rằng các kiến trúc hiện đại kém hiệu quả trong mọi bối cảnh. Một cách diễn giải thận trọng hơn là với CIFAR-10 32×32 và quy trình huấn luyện thống nhất, lợi thế của một số kiến trúc được thiết kế cho ảnh lớn chưa được phát huy tối đa. Việc tối ưu riêng siêu tham số, độ phân giải đầu vào và chiến lược chính quy hóa có thể làm thay đổi thứ hạng mô hình trong các nghiên cứu tiếp theo.

Từ góc độ ứng dụng, ResNet18-CIFAR phù hợp khi mục tiêu chính là tối đa hóa độ chính xác/F1-macro và có đủ tài nguyên tính toán. Ngược lại, SimpleCNN phù hợp hơn với bối cảnh cần mô hình đơn giản, dễ triển khai, dễ giải thích và có độ trễ thấp. Đây là khác biệt quan trọng giữa tối ưu hiệu năng dự đoán và tối ưu chi phí triển khai.

Tổng hợp lại, kết quả cho thấy không có một kiến trúc duy nhất tối ưu trên mọi tiêu chí. Việc lựa chọn mô hình cần dựa trên mục tiêu cụ thể: nếu ưu tiên độ chính xác thì ResNet18-CIFAR là lựa chọn tốt nhất trong thí nghiệm này; nếu ưu tiên dung lượng và độ trễ CPU thì SimpleCNN cân bằng hơn, còn nếu chỉ ưu tiên MACs lý thuyết thì MobileNetV2 là thấp nhất.



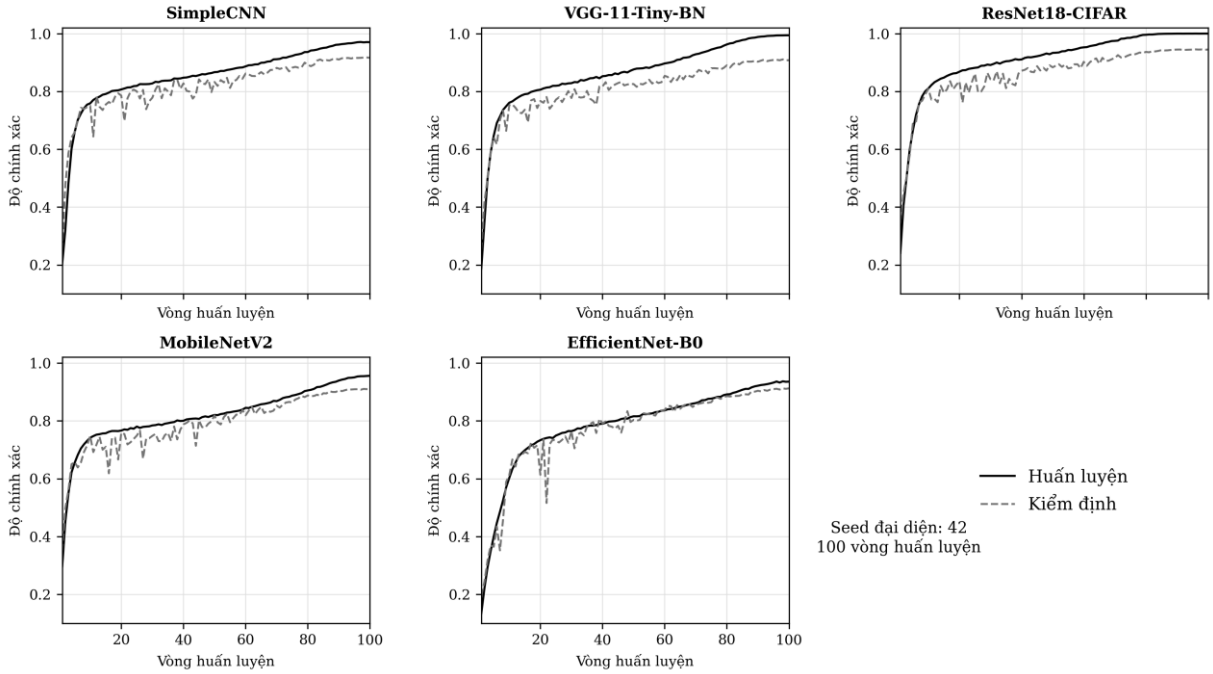
Hình 2. **So sánh độ chính xác và F1-macro của năm mô hình CNN trên CIFAR-10.**

Hình 2 cho thấy ResNet18-CIFAR vượt trội rõ rệt về độ chính xác và F1-macro, trong khi SimpleCNN, VGG-11-Tiny-BN và MobileNetV2 nằm trong nhóm hiệu năng trung bình khá gần nhau. Sự khác biệt này là cơ sở để tách riêng hai góc nhìn đánh giá: hiệu năng phân loại và chi phí triển khai.

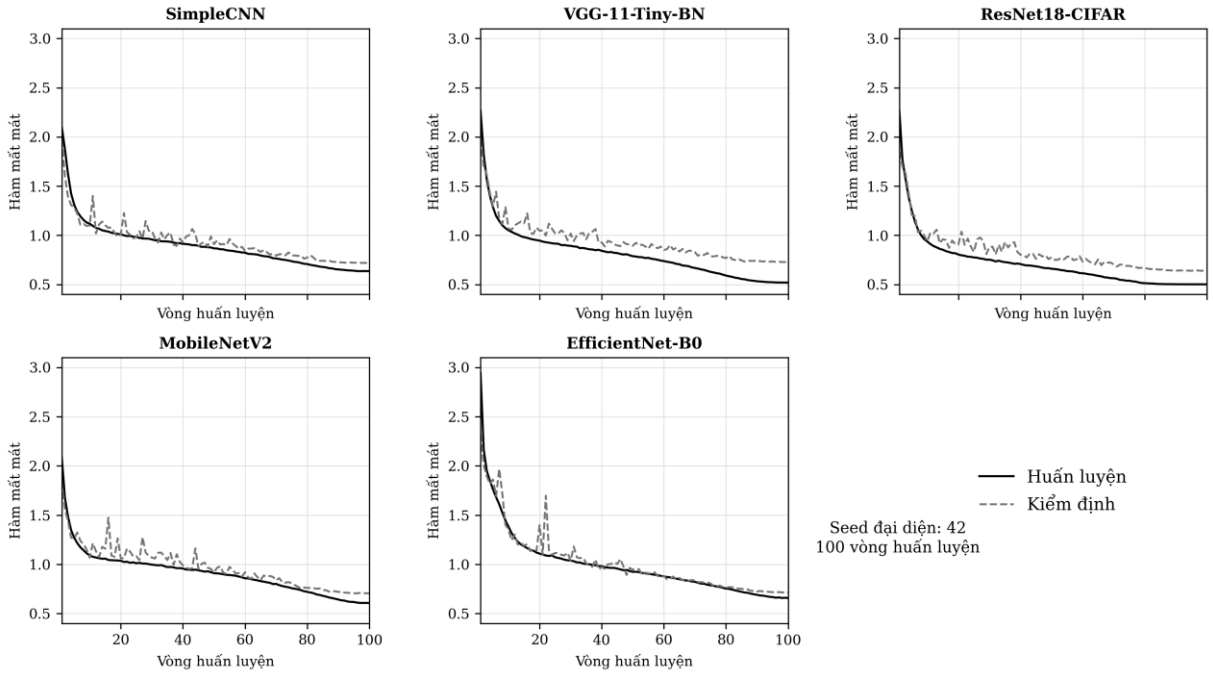
Khi kết hợp Hình 2 với Bảng 3, có thể thấy mô hình có độ chính xác cao nhất không phải là mô hình có chi phí thấp nhất. Vì vậy, bài báo tránh khẳng định một mô hình "vượt trội tuyệt đối", mà nhấn mạnh sự đánh đổi giữa độ chính xác, số tham số, FLOPs và độ trễ suy luận.

Để kiểm tra quá trình hội tụ, Hình 3 và Hình 4 trình bày lần lượt đường cong độ chính xác và hàm mất mát trên tập huấn luyện/kiểm định của seed 42. Việc sử dụng cùng một seed

đại diện cho cả năm mô hình giúp so sánh trực quan dưới cùng điều kiện khởi tạo; các kết luận định lượng cuối cùng vẫn dựa trên trung bình năm seed.



Hình 3. **Đường cong độ chính xác huấn luyện và kiểm định của năm mô hình trên** CIFAR-10 (seed 42).



Hình 4. **Đường cong hàm mất mát huấn luyện và kiểm định của năm mô hình trên CIFAR-10 (seed 42).**

DOI: https://doi.org/10.65153/5vswxq05

Các đường cong cho thấy cả năm mô hình đều cải thiện nhanh trong giai đoạn đầu và tiếp tục hội tụ khi tốc độ học giảm dần. Ở seed 42, độ chính xác kiểm định tốt nhất lần lượt là 91,72% (SimpleCNN, epoch 99), 91,06% (VGG-11-Tiny-BN, epoch 98), 94,52% (ResNet18-CIFAR, epoch 96), 91,04% (MobileNetV2, epoch 98) và 91,24% (EfficientNet-B0, epoch 100). ResNet18-CIFAR đạt mức kiểm định cao nhất nhưng khoảng cách train–validation ở cuối quá trình cũng lớn hơn, cho thấy dấu hiệu quá khớp nhất định. Các đường loss kiểm định ổn định ở giai đoạn cuối, không xuất hiện phân kỳ kéo dài; việc lựa chọn checkpoint tốt nhất theo tập kiểm định giúp hạn chế ảnh hưởng của dao động ở các epoch cuối.

3.2. Phân tích chi phí – hiệu quả

Bảng 4 trình bày hiệu năng phân loại của năm mô hình theo Accuracy và F1-macro, báo cáo dưới dạng trung bình ± độ lệch chuẩn qua năm lần chạy. ResNet18-CIFAR đạt kết quả cao nhất, còn SimpleCNN đứng sau nhưng có ưu thế rõ ràng về chi phí tính toán. Kết quả này củng cố quan điểm rằng việc đánh giá mô hình cần đặt trong quan hệ đa tiêu chí, thay vì chỉ xét một chỉ số đơn lẻ.

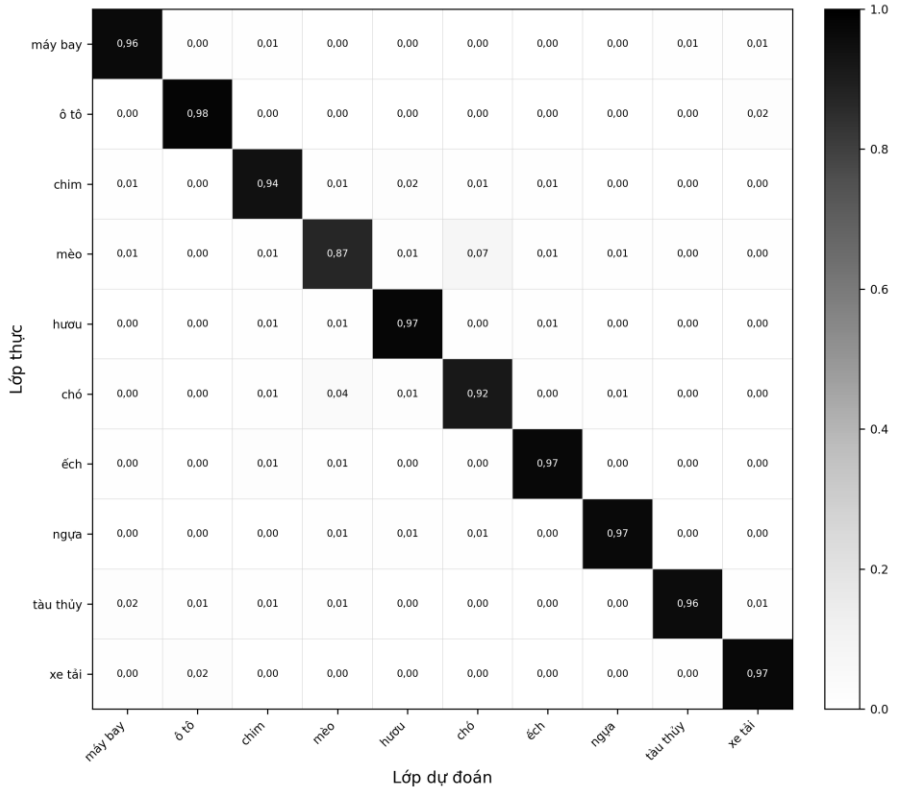
***Bảng 4*. Hiệu năng phân loại CIFAR-10 của năm mô hình CNN qua 5 lần chạy**

Mô hình	Số lần chạy	Độ chính xác (%)	F1-macro (%)
SimpleCNN	5	91,16 ± 0,08	91,15 ± 0,08
VGG-11-Tiny-BN	5	90,44 ± 0,30	90,43 ± 0,30
ResNet18-CIFAR	5	94,68 ± 0,23	94,68 ± 0,23
MobileNetV2	5	90,70 ± 0,12	90,67 ± 0,11
EfficientNet-B0	5	89,71 ± 0,25	89,69 ± 0,23

Ghi chú: Ký hiệu ± biểu thị độ lệch chuẩn qua 5 lần chạy với các hạt giống ngẫu nhiên khác nhau.

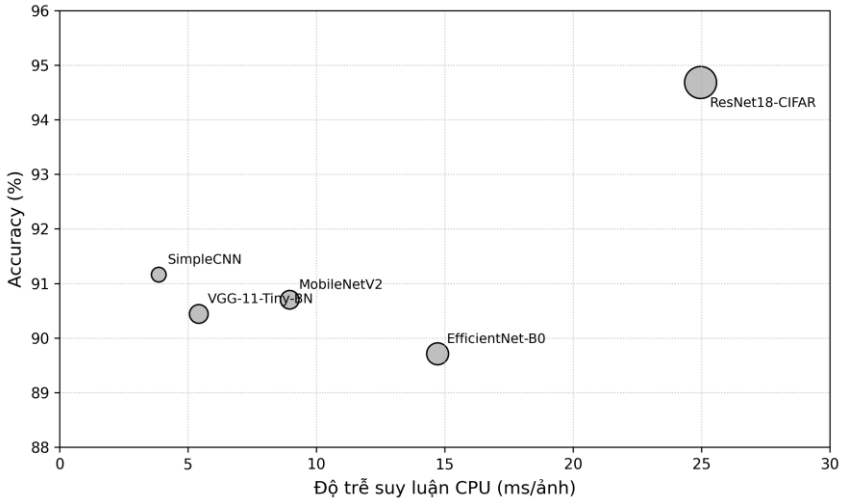
Kết quả ở Bảng 4 cho thấy ResNet18-CIFAR đạt độ chính xác và F1-macro cao nhất trong nhóm mô hình được khảo sát, trong khi SimpleCNN vẫn duy trì mức hiệu năng tương đối cao với chi phí tính toán thấp hơn đáng kể. Tuy nhiên, các chỉ số trung bình như độ chính xác và F1-macro chưa phản ánh đầy đủ mô hình thường nhầm lẫn ở những lớp nào. Do đó, để phân tích sâu hơn hành vi phân loại của mô hình, nghiên cứu tiếp tục xem xét ma trận nhầm lẫn trên tập kiểm thử CIFAR-10. Hình 5 trình bày ma trận nhầm lẫn chuẩn hóa của mô hình ResNet18-CIFAR, là mô hình đạt hiệu năng phân loại cao nhất trong các thí nghiệm.

DOI: https://doi.org/10.65153/5vswxq05



Hình 5. Ma trận nhầm lẫn chuẩn hóa theo hàng của mô hình ResNet18-CIFAR trên tập kiểm thử CIFAR-10.

Hình 5 cho thấy ResNet18-CIFAR nhận dạng tốt phần lớn các lớp trên CIFAR-10, thể hiện qua các giá trị cao trên đường chéo chính. Các giá trị ngoài đường chéo giúp nhận diện những cặp lớp dễ nhầm lẫn, ví dụ mèo–chó, chó–mèo, hươu–mèo hoặc máy bay–tàu thủy. Việc hiển thị đầy đủ các ô trong ma trận giúp người đọc quan sát rõ hơn phân bố sai số của mô hình thay vì chỉ nhìn các giá trị lớn.



Hình 6. Quan hệ giữa độ chính xác và độ trễ suy luận CPU của năm mô hình CNN trên CIFAR-10.

DOI: https://doi.org/10.65153/5vswxq05

Hình 6 minh họa sự đánh đổi giữa độ chính xác và độ trễ suy luận CPU. ResNet18-CIFAR đạt độ chính xác cao nhất nhưng nằm ở vùng có độ trễ lớn nhất. SimpleCNN có độ chính xác thấp hơn nhưng độ trễ thấp nhất, do đó phù hợp với các kịch bản ưu tiên tốc độ, chi phí và tính đơn giản. MobileNetV2 có FLOPs thấp nhưng độ trễ không thấp nhất, nhấn mạnh rằng độ trễ thực tế không thể được suy ra chỉ từ FLOPs.

4. TRAO ĐỔI VÀ KẾT LUẬN

Bài báo đã trình bày một nghiên cứu thực nghiệm so sánh năm kiến trúc CNN trên tập dữ liệu CIFAR-10 với quy trình huấn luyện thống nhất, gồm cùng tiền xử lý, tăng cường dữ liệu, bộ tối ưu, lịch điều chỉnh tốc độ học và năm hạt giống ngẫu nhiên. Hiệu năng được đánh giá đồng thời theo độ chính xác, F1-macro và các thước đo chi phí tính toán như số tham số, FLOPs và độ trễ suy luận trên CPU.

Kết quả mới cho thấy ResNet18-CIFAR đạt hiệu năng phân loại cao nhất với độ chính xác và F1-macro đều đạt 94,68 ± 0,23%. Tuy nhiên, mô hình này cũng có chi phí tính toán và độ trễ cao nhất. SimpleCNN đạt hiệu năng thấp hơn nhưng vẫn cạnh tranh, đồng thời có số tham số nhỏ nhất và độ trễ suy luận thấp nhất; riêng MobileNetV2 có MACs thấp nhất. Vì vậy, bài báo kết luận rằng ResNet18-CIFAR phù hợp khi ưu tiên độ chính xác, SimpleCNN phù hợp hơn khi ưu tiên dung lượng và độ trễ CPU, còn MobileNetV2 phù hợp khi MACs lý thuyết là ràng buộc chính.

Phân tích ma trận nhầm lẫn cho thấy mô hình tốt nhất vẫn còn nhầm lẫn ở một số lớp có đặc trưng thị giác gần nhau, đặc biệt trong nhóm ảnh động vật. Điều này gợi ý rằng để cải thiện thêm hiệu năng, các hướng như tăng cường dữ liệu chuyên biệt, chính quy hóa phù hợp, tối ưu siêu tham số riêng hoặc sử dụng chiến lược huấn luyện nâng cao có thể mang lại lợi ích.

Từ góc độ ứng dụng, kết quả nghiên cứu nhấn mạnh rằng lựa chọn mô hình cần gắn với mục tiêu sử dụng và ràng buộc tài nguyên. Trong môi trường giảng dạy, phòng thí nghiệm hoặc các hệ thống hạn chế phần cứng, một mô hình đơn giản như SimpleCNN vẫn có giá trị thực tiễn nhờ dễ cài đặt, dễ giải thích và độ trễ thấp. Trong khi đó, các kiến trúc sâu hơn như ResNet18-CIFAR phù hợp hơn khi mục tiêu chính là tối đa hóa độ chính xác.

Nghiên cứu còn một số hạn chế: mới khảo sát trên CIFAR-10, chưa tối ưu siêu tham số riêng cho từng kiến trúc, chưa thực hiện kiểm định thống kê sâu và mới đo độ trễ trên CPU trong môi trường Google Colab. Hướng nghiên cứu tiếp theo là mở rộng sang CIFAR-100,

Tiny-ImageNet, tối ưu cấu hình riêng, đo độ trễ trên GPU/thiết bị biên và khảo sát các kỹ thuật nén, lượng tử hóa hoặc chưng cất mô hình.

TÀI LIỆU THAM KHẢO

[1]. A. Krizhevsky (2009). Learning Multiple Layers of Features from Tiny Images, Technical Report, University of Toronto.

[2]. K. Simonyan and A. Zisserman (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition, International Conference on Learning Representations.

[3]. K. He, X. Zhang, S. Ren, and J. Sun (2016). Deep Residual Learning for Image Recognition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, tr. 770-778.

[4]. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, tr. 4510-4520.

[5]. M. Tan and Q. V. Le (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, Proceedings of the International Conference on Machine Learning, tr. 6105-6114.

[6]. I. Goodfellow, Y. Bengio, and A. Courville (2016). Deep Learning, MIT Press, Cambridge.

[7]. R. C. Gonzalez and R. E. Woods (2018). Digital Image Processing, 4th ed., Pearson, New York.

[8]. H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz (2018). mixup: Beyond Empirical Risk Minimization, International Conference on Learning Representations.

[9]. E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le (2019). AutoAugment: Learning Augmentation Policies from Data, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, tr. 113-123.

[10]. I. Loshchilov and F. Hutter (2019). Decoupled Weight Decay Regularization, International Conference on Learning Representations.

[11]. I. Loshchilov and F. Hutter (2017). SGDR: Stochastic Gradient Descent with Warm Restarts, International Conference on Learning Representations.