NHẬN DẠNG HÌNH ẢNH TỰ NHIÊN SỬ DỤNG MÔ HÌNH MẠNG NEURON TÍCH CHẬP

NATURAL IMAGE RECOGNITION BASED ON CONVOLUTIONAL NEURAL NETWORK

Vương Quang Phước[1], Hồ Phước Tiến[2]

[1]Trường Đại học Khoa học - Đại học Huế; vqphuoc@husc.edu.vn
[2]Trường Đại học Bách khoa - Đại học Đà Nẵng; hptien@dut.udn.vn

Tóm tắt - Gần đây, kỹ thuật Deep Learning đã tạo ra những bước tiến lớn trong việc giải quyết các bài toán về thị giác máy tính. Bằng cách sử dụng kiến trúc mạng neuron mới – mạng neuron tích chập (Convolutional Neural Network - CNN) –, ta có thể khắc phục được những trở ngại của mạng neuron truyền thống, tức dạng Perceptron đa lớp (Multilayer Perceptrons - MLP), và từ đó giúp việc huấn luyện mạng neuron hiệu quả hơn. Tuy nhiên, kiến trúc MLP cũng có những ưu điểm đối với việc xử lý cục bộ trong miền không gian. Bài báo trình bày một kiến trúc kết hợp giữa CNN và MLP để khai thác ưu điểm của hai kiến trúc này trong việc nhận dạng hình ảnh tự nhiên. Vai trò của các khối chức năng trong mạng sẽ được phân tích và đánh giá thông qua tỉ lệ nhận dạng. Việc đánh giá được thực hiện với bộ dữ liệu ảnh tự nhiên CIFAR-10. Quá trình thực nghiệm đã cho thấy những kết quả hứa hẹn về tỉ lệ nhận dạng, cũng như thể hiện được ưu điểm của kiến trúc kết hợp CNN và MLP.

Từ khóa - deep learning; neuron network; MLP; CNN; mô hình kết hợp; nhận dạng hình ảnh; CIFAR-10.

Abstract - Recently, Deep Learning has brought about interesting improvements in solving computer vision problems. By using a new specific architecture, i.e. Convolutional Neural Network (CNN), which has more advantages than the traditional one - known as Multilayer Perceptrons (MLP) -, we can improve performance of the training process. Yet, the MLP architecture is also useful for localized processing in the spatial domain. This paper considers an architecture combining both CNN and MLP to exploit their advantages for the problem of natural image recognition. The functional blocks in the network are analyzed and evaluated using recognition rate. The evaluation is carried out with a well-known dataset (CIFAR-10). The experiment shows promising results as well as benefits of a combination of the CNN and MLP architectures.

Key words - deep learning; neural network; MLP; CNN; combination of models; image recognition; CIFAR-10.

1. Đặt vấn đề

Deep Learning là một kỹ thuật của Machine Learning, cho phép huấn luyện mạng neuron nhiều lớp, cùng với một lượng dữ liệu lớn. Hiện nay, Deep Learning được ứng dụng nhiều trong các lĩnh vực thị giác máy tính, xử lý tiếng nói, hay xử lý ngôn ngữ tự nhiên với độ chính xác vượt trội so với các phương pháp truyền thống.

Nhìn chung, những kết quả của Deep Learning gắn liền với mạng CNN khi cho phép thực hiện mạng neuron nhiều lớp và khai thác mối quan hệ không gian (ví dụ với hình ảnh) [1, 2]. Gần đây, Lin [3] đã đề xuất ý tưởng kết hợp mạng CNN với MLP truyền thống, trong đó MLP cho phép khai thác thông tin cục bộ. Thật ra, MLP cũng có thể xem như là trường hợp riêng của CNN khi mà vùng kích thích (receptive field) có kích thước là 1x1. Để nhấn mạnh đặc điểm của cấu trúc này, sau đây nhóm tác giả vẫn sẽ sử dụng tên gọi MLP.

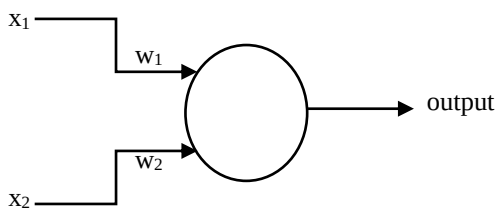
Bài báo này trình bày phương pháp giải quyết bài toán phân loại hình ảnh tự nhiên dựa trên mô hình kết hợp giữa mạng neuron tích chập (CNN) và mạng neuron truyền thống (MLP). Kiến trúc này giúp khai thác ưu điểm của mỗi kiểu mạng, nhằm nâng cao tỉ lệ nhận dạng ảnh. Ngoài ra, vai trò của số lượng khối con, tốc độ học (learning rate), cũng như cách loại bỏ ngẫu nhiên một số neuron trong mạng (dropout), hay quá trình tiền xử lý dữ liệu tác động đến kết quả nhận dạng sẽ được phân tích cụ thể trong phần thực nghiệm.

2. Mô hình kết hợp giữa MLP và CNN

Bản thân mỗi kiểu mạng MLP và CNN là một chủ đề lớn. Trong khuôn khổ giới hạn của bài báo, nhóm tác giả sẽ cố gắng trình bày những đặc điểm cơ bản về hai kiểu mạng neuron này, trước khi đi vào một kiến trúc kết hợp giữa CNN và MLP. Chi tiết về MLP và CNN có thể được tìm thấy ở [4].

2.1. Perceptron và Multi-layers Perceptron (MLP)

Một Perceptron có các ngõ vào nhị phân x_j và được gán tương ứng các trọng số w_j - thể hiện mức tác động của ngõ vào đến ngõ ra [4].



Hình 1. *Mô hình Perceptron đơn giản*

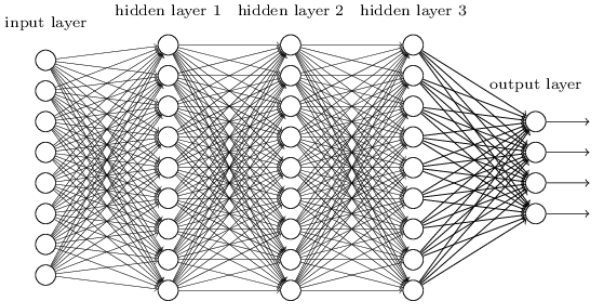
Ngõ ra sẽ được xác định là 1/0 phụ thuộc vào $\sum_j w_j x_j$ lớn/bé hơn giá trị ngưỡng threshold:

$$\text{output} = \begin{cases} 0 \text{ nếu } \sum_j w_j x_j \le \text{threshold} \\ 1 \text{ nếu } \sum_j w_j x_j > \text{threshold} \end{cases} \quad (1)$$

Tuy nhiên, Perceptron chỉ giải quyết được các bài toán tuyến tính đơn giản, mạng Perceptron đa lớp (MLP) được phát triển để giải các bài toán phức tạp hơn.

Cấu trúc MLP gồm một lớp đầu vào, một lớp đầu ra và một hay nhiều lớp neuron ẩn.

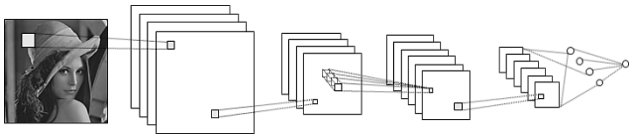
Các lớp ẩn sẽ làm nhiệm vụ tính toán và truyền thông tin từ ngõ vào đến ngõ ra, thông qua các kết nối đến toàn bộ node ở lớp phía trước và phía sau.



Hình 2. *Mô hình mạng MLP với 3 lớp ẩn [4]*

2.2. Mạng neuron tích chập (CNN)

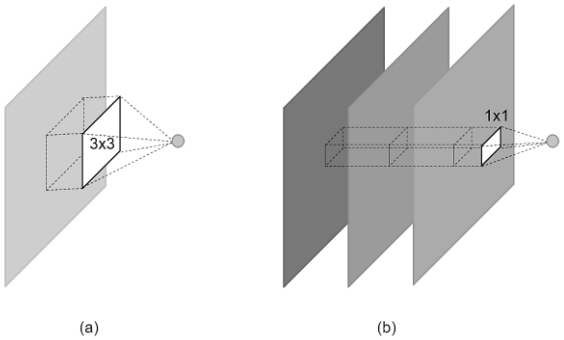
Về cơ bản, CNN bao gồm một vài lớp tích chập với các hàm kích hoạt phi tuyến áp vào đầu ra của lớp tích chập. Trong mạng MLP, mỗi neuron đầu vào được kết nối đến tất cả neuron của lớp kế tiếp. Ngược lại, ở mạng CNN, mỗi neuron trong một lớp chỉ liên kết với một số neuron lân cận với nó trong lớp kế trước. Lớp tích chập được thực hiện thông qua các bộ lọc: mỗi bộ lọc cho phép trích xuất một thuộc tính, như tần số, hướng; từ đó tạo nên bản đồ thuộc tính (feature map). Thông tin được lan truyền theo các lớp từ trước ra sau. Lớp cuối cùng thực hiện đánh giá để đưa ra quyết định ở ngõ ra. Một mô hình CNN cơ bản [1] được minh họa ở Hình 3.



Hình 3. *Minh họa mô hình cấu trúc mạng CNN Lenet:*
*Input → [Conv → Pool]*2 → FC → Output*

2.3. Kết hợp MLP và CNN

Với mạng CNN, các lớp tích chập cho phép khai thác thông tin trong miền không gian (ví dụ giữa các pixel lân cận nhau), nhưng không thực sự khai thác thông tin cục bộ (ví dụ, tại một pixel nhưng giữa các *feature map* khác nhau). Chính mạng MLP sẽ tập trung vào kiểu thông tin cục bộ này.

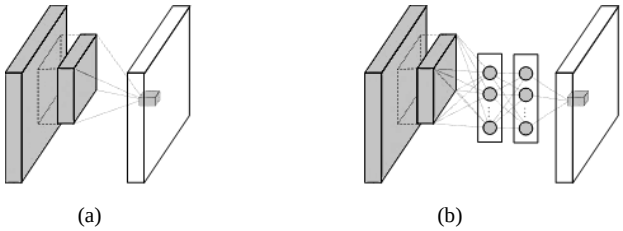


Hình 4. *(a) Khai thác thông tin theo miền không gian ở mô hình CNN, (b) Khai thác thông tin cục bộ ở mô hình MLP*

Hình 4 minh họa phương thức xử lý của CNN và MLP. Sau đây, ta sẽ xem xét các lớp chính trong kiến trúc kết hợp CNN-MLP, cũng như mô hình tổng thể của nó.

2.3.1. Lớp kết hợp CNN+MLP

Mạng MLP được thực hiện sau phép tích chập. Thực tế, thông qua nhiều bộ lọc, tích chập tạo ra nhiều feature map. Sau đó, MLP sẽ được áp dụng tại mỗi pixel nhưng với tất cả các thuộc tính (feature). Hình 5 minh họa lớp tích chập và sự kết hợp của lớp tích chập và MLP.

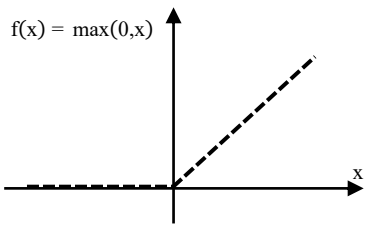


Hình 5. *(a) Lớp tích chập trong CNN;*
(b) Lớp tích chập kết hợp với MLP (vẽ lại theo [3])

2.3.2. Hàm kích hoạt phi tuyến

Mô hình sử dụng hàm ReLU (Rectified Linear Units Layers) để làm hàm kích hoạt phi tuyến. Mục đích của lớp này là để thêm thành phần phi tuyến cho mô hình. Các lớp ReLU được thêm vào vì những ưu điểm dễ thiết lập, tính toán nhanh và hiệu quả.

Công thức tính hàm ReLU:



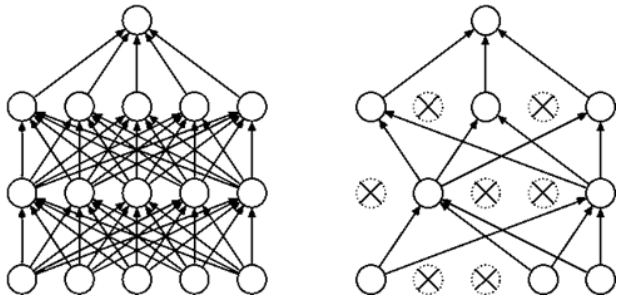
Hình 6. *Lớp ReLU chuyển đổi tất cả các giá trị âm về 0*

2.3.3. Pooling

Mục đích chính của các lớp Pooling [5] là để giảm kích thước dữ liệu, từ đó giảm tính toán trong mạng, đồng thời hạn chế overfitting – vấn đề này xảy ra khi mạng bám quá sát vào bộ dữ liệu huấn luyện. Pooling có thể xem như là phép lấy mẫu xuống. Mô hình được chọn sử dụng hai loại Pooling phổ biến hiện nay là Max Pooling - sử dụng cho các khối con, và Average Pooling - để xử lý thông tin toàn cục.

2.3.4. Dropout

Lớp Dropout nhằm giảm hiện tượng overfitting. Dropout loại bỏ một cách ngẫu nhiên một số neuron trong mạng bằng cách cho nó bằng 0 (bỏ kết nối). Có nghĩa là hệ thống sẽ quyết định ngõ ra trong khi thiếu thông tin. Lớp Dropout được đặc trưng bằng tỉ lệ dropout.



Hình 7. *Mạng neuron trước và sau quá trình Dropout, các node gạch chéo là các node đã bị loại bỏ [6]*

Ta sẽ xem xét ảnh hưởng của tỉ lệ này đến kết quả nhận dạng trong phần thực nghiệm. Quá trình loại bỏ ngẫu nhiên các node được minh họa trong Hình 7.

2.3.5. Hàm tổn hao

Hàm tổn hao Softmax đặt ở lớp cuối cùng trong mạng nhằm thực hiện giám sát quá trình huấn luyện mạng neuron. Hàm tổn hao sẽ so sánh kết quả dự đoán của mạng với nhãn thực sự đã có. Hàm có giá trị bé nếu kết quả dự đoán trùng với nhãn và ngược lại.

$$l(y,c) = -\log \frac{e^{y_c}}{\sum_{k=1}^{C} e^{y_k}} = -y_c + \log \sum_{k=1}^{C} e^{y_k} \quad (2)$$

Trong đó, y là véc-tơ đầu ra, C là số lượng nhãn, c là nhãn đã biết.

Quá trình huấn luyện nhằm cập nhật các trọng số để tối thiểu hóa hàm tổn hao. Trong mô hình này, cũng như trong các mô hình Deep Learning khác, kỹ thuật lan truyền ngược được sử dụng cho quá trình huấn luyện.

2.3.6. Mô hình kết hợp CNN-MLP

Mô hình kết hợp (gọi là CNN-MLP) được xây dựng dựa trên ý tưởng Network in Network [3]. Cấu trúc NiN được đề xuất gồm 3 khối con nối tiếp và 1 lớp Average Pooling.

Khối con hình thành dựa trên việc xen kẽ các cấu trúc CNN, MLP, các hàm kích hoạt phi tuyến ReLU, các lớp Max Pooling [5, 7], và cuối cùng là lớp Dropout để giảm hiện tượng overfitting. Các khối con này được sắp xếp liền kề nhau. Sau đó, thông tin được đưa vào lớp Average Pooling và lớp Softmax để quyết định giá trị ngõ ra.

Tiếp theo, ta sẽ khảo sát mô hình trên đây để làm rõ ưu điểm của mô hình kết hợp CNN-MLP so với mô hình CNN đơn thuần. Bên cạnh đó, một số yếu tố ảnh hưởng đến mô hình cũng sẽ được đánh giá.

3. Thực nghiệm và kết quả

Bài báo sử dụng bộ cơ sở dữ liệu ảnh tự nhiên CIFAR-10 [2] để đánh giá các mô hình. CIFAR-10, chứa 60.000 ảnh màu, được chia thành 10 nhóm, mỗi nhóm ứng với một loại đối tượng như máy bay, mèo, xe tải... Mỗi ảnh có kích thước 32x32. Ảnh trong CIFAR-10 có sự đa dạng về độ chiếu sáng, hướng, vị trí, tỉ lệ của các đối tượng.

Bộ dữ liệu này được chia thành hai phần: 50.000 ảnh dành cho huấn luyện, 10.000 ảnh còn lại dành cho kiểm tra. Sự phân chia này được dùng chung cho tất cả các mô hình được đánh giá trong bài báo này.

Hình 8. *Một số hình ảnh được chọn ngẫu nhiên từ bộ dữ liệu CIFAR-10 (kích thước 32x32)*

Tỉ lệ lỗi được chọn làm tiêu chí để đánh giá chất lượng nhận dạng của các mô hình, và được định nghĩa như sau:

$$\text{Tỉ lệ lỗi} = \frac{\text{Số lượng ảnh nhận dạng sai}}{\text{Tổng số ảnh được nhận dạng}} \times 100 \quad (3)$$

Ta thấy rằng:

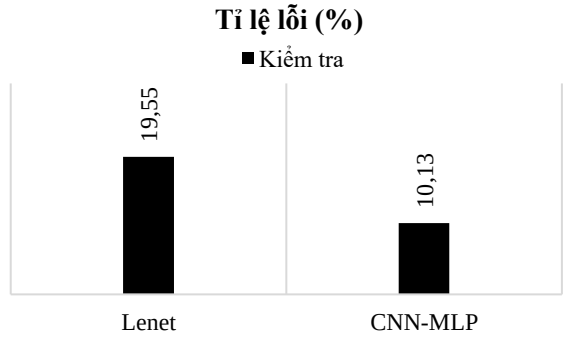
$$\text{Tỉ lệ nhận dạng đúng} = 100 - \text{tỉ lệ lỗi} \quad (4)$$

Quá trình thực nghiệm các mô hình được thực hiện dựa trên máy tính cá nhân, với cấu hình như sau: Intel Core i5-5200U 2.2GHz (4CPU), RAM 4GB, GPU NVIDIA Geforce 940M - VRAM 2GB. Thời gian huấn luyện mô hình xấp xỉ 18 giờ, ứng với 100 chu kỳ học (mỗi chu kỳ học/vòng lặp/epoch mất khoảng 11 phút để hoàn thành). MatConvNet [8] được dùng cho việc huấn luyện.

Sau đây ta sẽ đánh giá vai trò của MLP khi kết hợp với mạng CNN, cũng như ảnh hưởng của cấu trúc mạng đến kết quả nhận dạng.

3.1. So sánh cấu trúc CNN thuần và cấu trúc kết hợp

Để đánh giá chất lượng của mô hình mạng kết hợp trong việc nhận dạng hình ảnh tự nhiên, nhóm tác giả thực hiện so sánh kết quả mô hình LeNet (chỉ sử dụng CNN) và mô hình có kết hợp giữa CNN và MLP.



Hình 9. *So sánh tỉ lệ lỗi giữa mô hình LeNet và CNN-MLP*

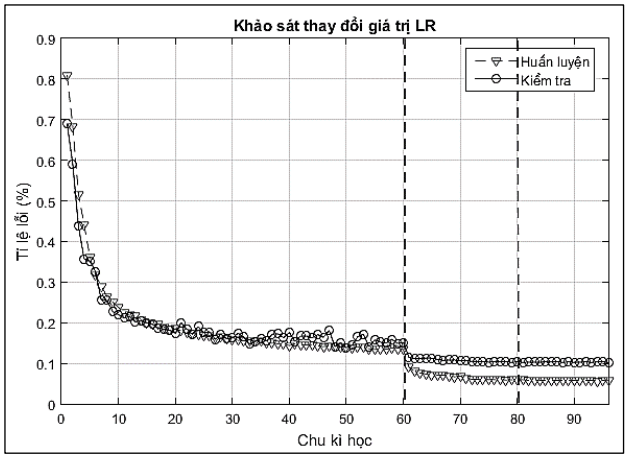
Hình 9 thể hiện tỉ lệ lỗi về nhận dạng ảnh trên tập kiểm tra của bộ dữ liệu CIFAR-10 đối với 2 mô hình LeNet và CNN-MLP. Trong đánh giá này, LeNet được giữ nguyên theo thiết kế ở [1] và được minh họa tại Hình 3; mô hình CNN-MLP sử dụng cấu trúc với 3 khối con, theo ý tưởng ở [3]. Cả hai mô hình này đều được huấn luyện cho đến khi tỉ lệ nhận dạng đúng không còn thay đổi đáng kể trong các chu kỳ học liên tiếp (sau 100 vòng lặp). Kết quả huấn luyện và kiểm tra cho thấy, tỉ lệ lỗi giảm từ 19,55% (LeNet) xuống còn 10,13% (CNN-MLP). Kết quả này chứng tỏ rằng, quá trình kết hợp thêm MLP đã góp phần nâng cao chất lượng nhận dạng so với việc chỉ sử dụng CNN đơn thuần.

Trên thực tế, khác biệt chính giữa mạng LeNet và CNN-MLP nằm ở chỗ không có và có MLP; trong khi cách xử lý trong khối CNN tương đối giống nhau giữa hai mạng này. Như vậy, khi được thực hiện sau phép tích chập, MLP tăng cường thêm phần xử lý cục bộ (tại mỗi vị trí không gian, nhưng trên các thuộc tính khác nhau – minh họa tại Hình 4), và có thể điều này đã cho phép trích ra những thuộc tính hữu ích hơn cho việc nhận dạng.

3.2. Tác động của tốc độ học – Learning Rate (LR)

Việc lựa chọn giá trị LR ảnh hưởng đến sự thay đổi tốc độ học của mô hình (sự thay đổi của các trọng số). Việc sử dụng LR thích hợp sẽ giúp rút ngắn được quá trình huấn luyện của mạng. Ở đây, nhóm tác giả sử dụng giá trị LR lớn cho các chu kỳ học đầu, giúp quá trình học nhanh hơn, sau đó thực hiện giảm LR đi 10 lần cho các chu kỳ học sau để quá trình tinh chỉnh các giá trị huấn luyện tốt hơn.

Hình 10 mô tả kết quả nhận dạng khi thay đổi LR, quá trình thay đổi được thực hiện tại vòng lặp (chu kỳ học) thứ 61 và vòng lặp thứ 81.



Hình 10. *Tác động của learning rate đến kết quả nhận dạng*

Qua kết quả được đưa ra ở Hình 10, có thể dễ dàng nhận thấy một số đặc điểm thú vị sau. Đầu tiên, kết quả kiểm tra gần như chỉ dao động xung quanh mức tỉ lệ lỗi 15% trước khi có sự thay đổi về learning rate xảy ra. Thứ hai, việc thay đổi tỉ lệ LR đã tạo ra bước nhảy vọt về tỉ lệ nhận dạng lỗi trong cả quá hình huấn luyện lẫn kiểm tra như kết quả tại vòng lặp thứ 61 (tỉ lệ lỗi trên tập kiểm tra hạ xuống ~11%). Cuối cùng, tại lần giảm giá trị LR ở vòng lặp thứ 81 thì tỉ lệ nhận dạng không có sự thay đổi rõ rệt, mô hình gần như đạt đến trạng thái giới hạn và tỉ lệ nhận dạng đúng đạt ngưỡng gần tối đa.

3.3. Vai trò của Dropout

Như đã đề cập ở phần trên, các lớp Dropout loại bỏ một số ngẫu nhiên các neuron, từ đó giúp cho quá trình huấn luyện không bị overfitting. Ta đánh giá vai trò của lớp Dropout trong việc nhận dạng hình ảnh thông qua việc sử dụng các tỉ lệ dropout khác nhau: 0% (tức không sử dụng lớp dropout), 30%, 50%, 70% và 90%. Chú ý rằng tỉ lệ dropout thể hiện tỉ lệ neuron được loại bỏ.

Với nội dung khảo sát khá nhiều, nhóm tác giả chỉ thực hiện khảo sát trên mô hình có 3 khối con (với các giá trị Dropout khác nhau) và ứng với mỗi mô hình, thực hiện huấn luyện trong 60 chu kỳ học. Kết quả đưa ra thể hiện xu thế học của mô hình và được mô tả ở Bảng 1.

Bảng 1. *Tỉ lệ lỗi khi thay đổi giá trị Dropout của hệ thống*

	0%	**30%**	**50%**	**70%**	**90%**
Huấn luyện	9,31	11,35	13,43	17,1	27,7
Kiểm tra	14,49	14,19	15,16	16,59	23,47

Thông qua kết quả nhận được, ta thấy mô hình không có lớp Dropout cho tỉ lệ lỗi khi huấn luyện thấp nhất (9,31%). Nhưng độ chênh lệch tỉ lệ lỗi giữa quá trình huấn luyện và kiểm tra lại cao hơn so với các trường hợp khác.

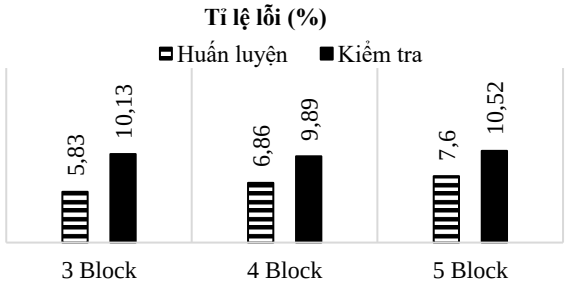
Mối quan hệ giữa tỉ lệ dropout, tỉ lệ lỗi khi huấn luyện và kiểm tra cho thấy được hiện tượng overfitting rõ ràng ở trường hợp không có lớp Dropout. Hiện tượng này giảm dần khi tăng tỉ lệ dropout. Tuy nhiên, khi tỉ lệ dropout quá lớn, ví dụ 90%, thì tỉ lệ lỗi nhận dạng trong huấn luyện và kiểm tra đều tăng vọt (thể hiện quá trình underfitting). Nguyên nhân gây ra hiện tượng này là thông tin bị mất khá nhiều trong quá trình xử lý dẫn đến phân loại không được chính xác.

Thực tế, để có được tỉ lệ phân loại tối ưu nhất, yêu cầu phải thực hiện thử nghiệm nhiều tỉ lệ khác nhau và riêng lẻ cho từng mô hình.

3.4. Ảnh hưởng của số lượng khối con trong mô hình kết hợp CNN-MLP

Cho đến nay, giải thích chặt chẽ sự hoạt động của mô hình mạng neuron nhiều lớp (Deep Learning), ví dụ bằng cách sử dụng mô hình toán học, vẫn còn là câu hỏi mở. Nói chung, cấu trúc mạng neuron được lựa chọn bằng cách dựa trên kết quả đầu ra, ứng với một công việc cụ thể. Theo cách tiếp cận trên, bài báo này cũng xem xét ảnh hưởng của số lượng các khối con lên khả năng nhận dạng của mạng kết hợp CNN-MLP.

Chú ý rằng, mỗi khối con chính là khối kết hợp chứa cả CNN, MLP, ReLU, Pooling, và Dropout.



Hình 11. *Tỉ lệ lỗi khi thay đổi số lượng khối con trong mô hình kết hợp*

Giữ nguyên cấu trúc các khối con, tăng số lượng các khối con từ 3 khối lên 4 khối và 5 khối, kết quả nhận được như Hình 11 (sau 100 vòng lặp). Trong trường hợp tăng số lượng các khối, tỉ lệ lỗi khi huấn luyện (màu xanh) tăng lên trong khi giá trị tỉ lệ lỗi khi kiểm tra không thay đổi quá lớn, cho thấy đã hạn chế được hiện tượng overfitting. Tuy nhiên, ở trường hợp sử dụng 4 khối con, tỉ lệ lỗi trong quá trình kiểm tra lại tốt hơn (10,13% → 9,89%). Và khi tăng số khối lên 4 hoặc 5, độ chênh lệch tỉ lệ nhận dạng sai trong quá trình kiểm tra và huấn luyện có xu hướng hạ xuống so với 2 trường hợp trước.

Như vậy, ở đây ta có thể có hai nhận xét sau. Thứ nhất, tăng số lượng khối con có thể làm tăng tỉ lệ nhận dạng đúng, ví dụ trường hợp dùng 4 khối con - với tỉ lệ nhận dạng đúng lên đến 90,11% (tỉ lệ lỗi 9,89%); tức là mạng nhiều lớp hơn có khả năng cao hơn trong việc học được những thuộc tính quan trọng và cần thiết cho quá trình nhận dạng. Thứ hai, khi tăng số lượng khối con thì độ chênh lệch giữa tỉ lệ lỗi trên tập kiểm tra so với tập huấn luyện có xu hướng giảm. Điều này chứng tỏ mạng có xu thế hạn chế hiện tượng overfitting khi số lượng khối con tăng lên. Và còn có thể tăng thời gian huấn luyện để cải thiện kết quả.

Trong bài báo này, do giới hạn về tài nguyên phần cứng nên nhóm tác giả chỉ dừng lại ở mô hình với 5 khối con. Có thể kết quả sẽ được cải thiện tốt hơn khi tăng số lượng khối con và tăng thời gian huấn luyện.

3.5. Tác động chuẩn hóa dữ liệu và whitening

Ngoài việc khảo sát mô hình mạng kết hợp dựa trên các siêu tham số, tác động của việc chuẩn hóa dữ liệu và whitening cũng được đánh giá. Mô hình sử dụng đánh giá là mô hình kết hợp đơn giản, với 3 khối con CNN-MLP

như trong một số khảo sát đã thực hiện. Kết quả nhận được sau khi thực hiện 45 vòng lặp huấn luyện, lựa chọn với tỉ lệ dropout 0,5, áp dụng giảm learning rate tại vòng lặp thứ 40. Bảng 2 thể hiện kết quả khảo sát được trong trường hợp có và không có tiền xử lý dữ liệu.

Bảng 2. *Kết quả nhận dạng trong trường hợp có và không có tiền xử lý dữ liệu*

Mô hình		Tỉ lệ kiểm tra lỗi (%)
1	Thực hiện tiền xử lý dữ liệu (chuẩn hóa + whitening)	11,61
2	Không thực hiện chuẩn hóa	12,26
3	Không thực hiện whitening	15,95
4	Không thực hiện tiền xử lý	15,36

Dựa vào các kết quả trên ta nhận thấy được kết quả nhận dạng của mô hình (1) là tốt nhất, và giảm dần theo thứ tự (2), (4), (3).

Quá trình chuẩn hóa dữ liệu có tác động đến kết quả phân loại, tuy nhiên ảnh hưởng không lớn (chênh lệch ~0,6% - xét trên tỉ lệ nhận dạng sai của mô hình (1) và (2)) đối với bộ dữ liệu CIFAR10.

So sánh kết quả của mô hình (1) và (3), ta thấy tỉ lệ nhận dạng lỗi tăng vọt từ 11,61% lên 15,95%. Khác biệt giữa hai mô hình này là có và không có xử lý whitening. Như vậy, có thể thấy rằng bằng cách hạn chế sự tương quan giữa các phần tử trong ảnh, xử lý whitening có tác động lớn đến kết quả nhận dữ liệu, giúp tăng được tỉ lệ nhận dạng đúng của hệ thống. Ở mô hình (4), loại bỏ cả hai quá trình chuẩn hóa và whitening, tỉ lệ nhận dạng đúng thấp hơn so với mô hình (1) và (2). Và với mô hình (3) thì lại có kết quả tương đồng vì mức độ ảnh hưởng của chuẩn hóa dữ liệu trong các khảo sát này không cao.

Các kết quả thực nghiệm trên cho thấy việc xử lý chuẩn hóa (tuy kết quả chưa nhận thấy quá rõ ràng) và whitening giúp tăng tỉ lệ nhận dạng ảnh, và đáng cân nhắc để đưa vào mô hình.

3.6. Ảnh hưởng của Batch size đến kết quả nhận dạng

Batch size quyết định số lượng ảnh được dùng cho mỗi lần cập nhật trọng số, ví dụ, với kích thước tập huấn luyện của CIFAR-10 là 50.000 ảnh. Một chu kỳ học tương ứng với xử lý 50.000 ảnh. Nếu chọn batch size = 20, có nghĩa là dữ liệu sẽ được chia thành 2.500 gói con để xử lý. Tương tự, nếu batch size = 200, thì số gói dữ liệu là 250 gói. Bảng 3 là các kết quả khảo sát thu được sau 60 vòng lặp trên mô hình 3 khối con.

Bảng 3 là kết quả khi có sự thay đổi về kích thước gói dữ liệu batch. Với kích thước Batch size nhỏ, tỉ lệ nhận dạng lỗi cao (47,41%). Khi kích thước tăng dần, kết quả khảo sát đã có những thay đổi khả quan. Nhìn chung, khi batch size lớn, các trọng số sẽ được cập nhật một cách ổn định hơn. Nhưng cũng lưu ý rằng, batch size lớn sẽ yêu cầu nhiều bộ nhớ hơn.

Bảng 3. *Kết quả nhận dạng khi sử dụng dữ liệu với các batch size khác nhau trên cùng một mô hình mạng*

Batch size	10	20	50	100	200
Tỉ lệ lỗi (%)	47,41	22,88	17,43	15,16	14,45

4. Kết luận

Bài báo đã thực hiện nhận dạng ảnh tự nhiên dựa trên sự kết hợp giữa mạng neuron tích chập CNN và mạng perceptron đa lớp MLP. Trong đó, MLP được sử dụng để khai thác thêm thông tin cục bộ, bên cạnh thông tin về mặt không gian từ CNN. Kết quả thực nghiệm cho thấy sự kết hợp CNN-MLP cho phép cải thiện tỉ lệ nhận dạng. Ngoài ra, bài báo cũng phân tích tác động của tốc độ học đến việc rút ngắn thời gian huấn luyện, cho thấy vai trò của lớp dropout trong việc giảm overfitting, tầm quan trọng của quá trình tiền xử lý dữ liệu, cũng như kích thước gói batch size ảnh hưởng đến kết quả nhận dạng. Khi tăng độ sâu của mạng, ta nhận thấy xu hướng cải thiện chất lượng nhận dạng. Xu hướng này có thể càng được thể hiện rõ khi thời gian huấn luyện càng lớn.

Bài báo hiện tại quan tâm đến nhận dạng ảnh với kích thước nhỏ. Bằng cách kết hợp với phương pháp phát hiện sự nổi bật [9], mô hình trên có thể sẽ giúp nhận dạng đối tượng trong bối cảnh thực tế hơn.

TÀI LIỆU THAM KHẢO

[1] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based Learning Applied to Document Recognition", *Proceedings of The IEEE 86 (11)*, 1998, pp. 2278-2324.

[2] A. Krizhevsky, G. Hinton, *Learning Multiple Layers of Features from Tiny Images*, Technical report, University of Toronto, 2009.

[3] Min Lin, Qiang Chen, Shuicheng Yan, *Network in Network*, arXiv:1312.4400v3, 2014.

[4] Michael A. Nielsen, *Neural Networks and Deep Learning*, Determination Press, 2015.

[5] Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, MIT Press, 2016.

[6] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", *Journal of Machine Learning Research*, 2014, pp. 1929-1958.

[7] Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, Yoshua Bengio, *MaxOut Network*, arXiv:1302.4389v4, 2013.

[8] Andrea Vedaldi, Karel Lenc, *MatConvNet - Convolutional Neural Networks for MATLAB*, arXiv:1412.4564, 2016.

[9] T. Ho-Phuoc, *Développement et mise en oeuvre de modèles d'attention visuelle*, PhD thesis, Université Joseph Fourier, 2010.

(BBT nhận bài: 13/9/2017, hoàn tất thủ tục phản biện: 16/5/2018)