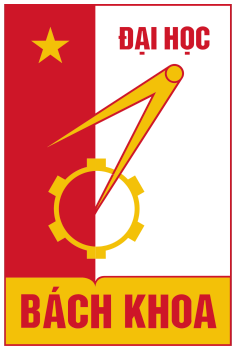
ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO PROJECT I

PHÂN LOẠI HÌNH ẢNH SỬ DỤNG CÁC PHƯƠNG PHÁP HỌC MÁY VÀ HỌC SÂU

Đàm Phú Đạt

MSSV: 20235029

Ngành: Khoa Học Máy Tính

Giảng viên hướng dẫn: CN. Trần Thị Lan Anh

Lớp: 755566

Hà Nội, 01/2026

Mục lục

1 Giới thiệu **2**

2 Bài toán và Dữ liệu **2**

2.1 Bài toán phân loại ảnh 2

2.2 Dữ liệu 3

3 Cơ sở lý thuyết **4**

3.1 Các phương pháp học máy 4

3.1.1 Softmax Regression 4

3.1.2 Mạng neural truyền thẳng 5

3.2 Mạng neural tích chập 7

3.2.1 Tổng quan 7

3.2.2 AlexNet 8

3.2.3 ResNet 8

3.3 Vision Transformer 10

3.3.1 Transformer 10

3.3.2 Vision Transformer 11

4 Triển khai **12**

4.1 Tiền xử lý dữ liệu 12

4.2 Triển khai 13

4.3 Tiêu chí đánh giá 14

5 Kết quả thực nghiệm **15**

6 Kết luận và Hướng phát triển **17**

1 Giới thiệu

Thị giác máy tính là một trong những lĩnh vực quan trọng trong trí tuệ nhân tạo. Trong đó, phân loại hình ảnh là bài toán nền tảng với nhiều ứng dụng thực tiễn, từ nhận diện đối tượng, giám sát, tới y tế và robot. Các phương pháp học máy (Machine Learning) và học sâu (Deep Learning) đã mang lại những bước tiến đáng kể trong việc giải quyết bài toán phân loại ảnh, mở ra nhiều hướng tiếp cận khác nhau, đi kèm với những ưu điểm và hạn chế riêng về độ chính xác, khả năng tổng quát hóa và chi phí tính toán.

Trong bối cảnh đó, dự án này được thực hiện với mục tiêu tìm hiểu và so sánh một số phương pháp phân loại ảnh tiêu biểu thông qua thực nghiệm trên bộ dữ liệu chuẩn CIFAR-10 [1]. Các mô hình được lựa chọn bao gồm Softmax Regression, mạng neuron truyền thẳng, AlexNet [2], ResNet [3] và Vision Transformer [4]. Đây là những mô hình đại diện cho các giai đoạn phát triển khác nhau của lĩnh vực phân loại ảnh, từ các phương pháp học máy truyền thống đến các kiến trúc học sâu tiên tiến.

Thông qua việc triển khai, huấn luyện và đánh giá các mô hình trên cùng một bộ dữ liệu, dự án hướng tới việc phân tích sự khác biệt về hiệu năng phân loại, tốc độ huấn luyện và chi phí tính toán của từng phương pháp. Kết quả so sánh sẽ giúp rút ra những nhận xét về ưu nhược điểm của từng kỹ thuật, từ đó định hướng cho việc lựa chọn mô hình cho các bài toán phân loại ảnh trong thực tế. Mã nguồn đầy đủ và các tài liệu bổ sung của dự án được công bố công khai [1].

Phần còn lại của báo cáo này được tổ chức như sau.

Chương 2 trình bày về bài toán phân loại ảnh và bộ dữ liệu được sử dụng.

Chương 3 trình bày cơ sở lý thuyết về các mô hình được sử dụng.

Chương 4 mô tả chi tiết quy trình thực nghiệm, bao gồm chuẩn bị dữ liệu, thiết lập mô hình, các tham số huấn luyện và tiêu chí đánh giá

Chương 5 trình bày kết quả thực nghiệm, kèm theo phân tích, nhận xét và so sánh giữa các mô hình.

Cuối cùng, Chương 6 tổng hợp các kết quả chính và đề xuất một số hướng phát triển trong tương lai.

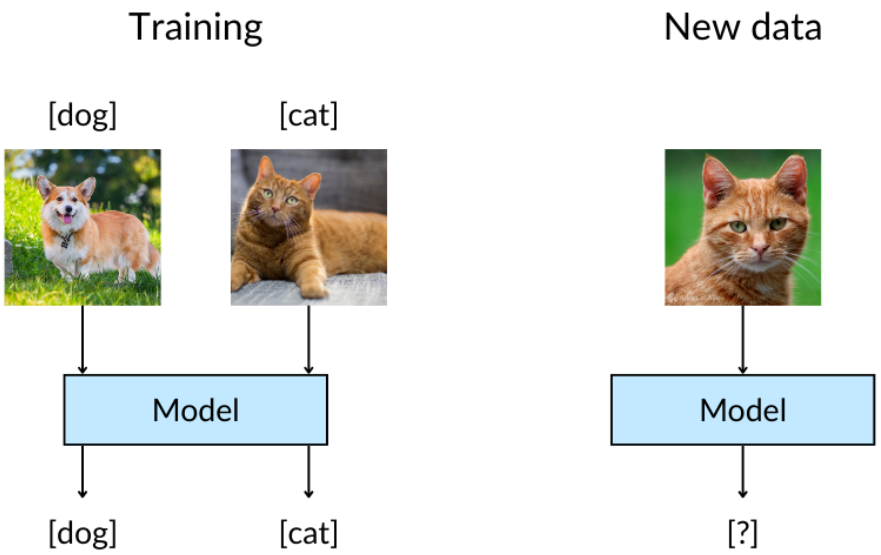
2 Bài toán và Dữ liệu

2.1 Bài toán phân loại ảnh

Phân loại ảnh là một trong những bài toán nền tảng của thị giác máy tính. Đó là bài toán học có giám sát, trong đó một mô hình học máy được huấn luyện với các cặp (ảnh, nhãn) và cần học cách gán nhãn cho một ảnh đầu vào, như mình họa trong Hình 1. Một

[1] `https://github.com/phudatdam/it3150-image-classification`

đặc điểm quan trọng của dữ liệu ảnh là tính không gian: các pixel không tồn tại độc lập mà có quan hệ lân cận với nhau, tạo nên cấu trúc hình dạng, hoa văn và vật thể. Ngoài ra, ảnh có thể thay đổi theo điều kiện ánh sáng, góc chụp, độ phân giải, tỉ lệ phóng to, thu nhỏ...



Hình 1: Minh họa bài toán phân loại ảnh trong giai đoạn huấn luyện và dự đoán dữ liệu mới

Mục tiêu của bài toán phân loại là học một hàm ánh xạ $f : \mathbb{R}^{H \times W \times C} \to \{1, 2, ..., K\}$, trong đó ảnh đầu vào là tensor ba chiều và mô hình cần dự đoán đúng nhãn lớp tương ứng trong số K lớp đầu ra.

2.2 Dữ liệu

Bộ dữ liệu được sử dụng trong dự án là CIFAR-10 [1]. CIFAR-10 bao gồm tổng cộng 60,000 ảnh màu (RGB), mỗi ảnh có kích thước 32×32 pixel, được phân chia đều vào 10 lớp đối tượng khác nhau, bao gồm: *airplane, automobile, bird, cat, deer, dog, frog, horse, ship* và *truck*, trong đó mỗi lớp có 6,000 ảnh (Hình 2).

Hình 2: Mẫu ảnh cho từng lớp trong bộ dữ liệu CIFAR-10

CIFAR-10 được lựa chọn vì nhiều lý do. Thứ nhất, bộ dữ liệu có kích thước vừa phải, kích thước ảnh nhỏ và số lớp tương đối hạn chế, phù hợp với điều kiện tính toán, nhưng đồng thời vẫn đủ đa dạng để thể hiện sự khác biệt giữa các phương pháp. Thứ hai, CIFAR-10 là một bộ dữ liệu tiêu chuẩn, được sử dụng phổ biến trong nghiên cứu, do đó kết quả có thể dễ dàng so sánh với các nghiên cứu liên quan. Ngoài ra, dữ liệu phân bố đồng đều giữa các lớp giúp hạn chế hiện tượng mất cân bằng dữ liệu và thuận lợi cho việc huấn luyện cũng như đánh giá các mô hình phân loại.

3 Cơ sở lý thuyết

Dựa trên đặc điểm bài toán phân loại ảnh đa lớp và bộ dữ liệu CIFAR-10 đã được trình bày ở chương trước, chương này tập trung giới thiệu cơ sở lý thuyết của phương pháp được sử dụng trong dự án. Mục tiêu của chương không chỉ nhằm trình bày nguyên lý hoạt động của từng mô hình, mà còn làm rõ lý do lựa chọn các phương pháp này để giải quyết bài toán phân loại ảnh trong bối cảnh dữ liệu thực nghiệm.

Trong chương này, mỗi phương pháp sẽ được trình bày về mặt lý thuyết, tập trung vào các ý tưởng cốt lõi liên quan trực tiếp đến bài toán. Các chi tiết triển khai cụ thể và thiết lập thực nghiệm sẽ được trình bày trong các chương tiếp theo.

3.1 Các phương pháp học máy

3.1.1 Softmax Regression

Softmax Regression là mô hình học máy cơ bản cho các bài toán phân loại nhiều lớp. Đầu vào của mô hình là một vector, và đầu ra là xác suất vector đó thuộc về mỗi lớp.

Giả sử $\mathbf{x} \in \mathbb{R}^d$ là vector đặc trưng đầu vào và K là số lớp cần phân loại. Mỗi lớp k có một vector trọng số $\mathbf{w}_k$ và một hệ số chệch b_k. Xác suất vector x thuộc vào lớp k là:

$$p(y = k \mid \mathbf{x}) = \frac{\exp(\mathbf{w}_k^\top \mathbf{x} + b_k)}{\sum_{j=1}^{K} \exp(\mathbf{w}_j^\top \mathbf{x} + b_j)}$$

Mục tiêu huấn luyện mô hình là cực tiểu hóa hàm mất mát cross-entropy:

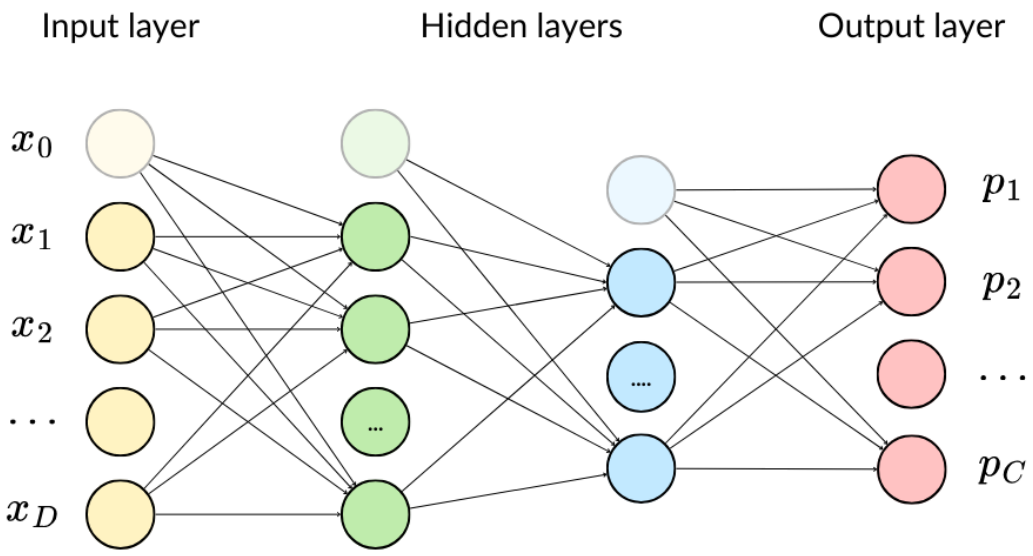
$$\mathcal{L} = -\sum_i \log p(y^{(i)} \mid \mathbf{x}^{(i)})$$

Hàm mất mát này khuyến khích mô hình gán xác suất cao cho lớp đúng của từng mẫu dữ liệu. Softmax Regression có ưu điểm là huấn luyện nhanh, dễ tối ưu và kết quả dễ diễn giải. Tuy nhiên, điểm hạn chế là mô hình chỉ có thể học các ranh giới tuyến tính. Điều này khiến Softmax Regression kém hiệu quả đối với dữ liệu có cấu trúc phức tạp như ảnh, khi mối quan hệ giữa các pixel mang tính phi tuyến cao.

Trong phân loại ảnh, Softmax Regression thường không được áp dụng trực tiếp, mà thường được dùng như lớp phân loại cuối cùng trên các đặc trưng đã trích xuất, hoặc dùng làm cơ sở để đánh giá mức độ khó của bài toán.

3.1.2 Mạng neural truyền thẳng

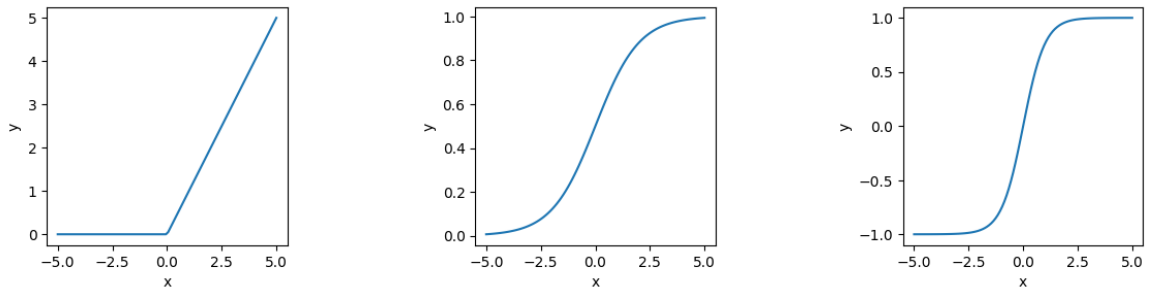
Mạng neural truyền thẳng (Fully Connected Neural Network - FCNN) bao gồm các neuron được tổ chức thành các tầng và tín hiệu được truyền theo một chiều từ đầu vào đến đầu ra (Hình 3).



Hình 3: Sơ đồ kiến trúc mạng neural truyền thẳng

Đầu ra của mỗi tầng là đầu vào của tầng kế tiếp. Một FCNN bao gồm tầng đầu vào,

các tầng ẩn và tầng đầu ra. Mỗi tầng ẩn gồm nhiều neuron, mỗi neuron thực hiện một phép tuyến tính $z = \mathbf{w}^\top \mathbf{x} + b$, sau đó đi qua một hàm kích hoạt phi tuyến $a = \sigma(z)$. Các hàm kích hoạt phổ biến gồm ReLU, sigmoid và tanh (Hình 4), trong đó ReLU được sử dụng rộng rãi ở các tầng ẩn. Hàm kích hoạt ở tầng đầu ra tùy thuộc vào yêu cầu bài toán, thường là softmax với phân loại nhiều lớp. Nhờ kết hợp nhiều lớp phi tuyến, FCNN có khả năng biểu diễn các hàm số phức tạp, vượt trội so với các mô hình tuyến tính.



(a) ReLU (b) sigmoid (c) tanh

Hình 4: Đồ thị một số hàm kích hoạt phổ biến

FCNN được huấn luyện bằng thuật toán lan truyền gradient ngược (backpropagation) kết hợp với các kỹ thuật tối ưu như Gradient Descent. Việc điều chỉnh trọng số qua nhiều vòng huấn luyện cho phép mạng dần cải thiện khả năng phân loại dựa trên dữ liệu đầu vào.

Các mô hình tuyến tính như Softmax Regression và FCNN có đặc điểm chung là nhận đầu vào là vector. Đối với dữ liệu hình ảnh, ảnh đầu vào được trải phẳng thành một vector trước khi đưa vào mô hình. Cách biểu diễn này cho phép áp dụng trực tiếp các mô hình học máy cơ bản lên dữ liệu ảnh mà không cần các bước xử lý phức tạp.

Một số nghiên cứu về phân loại ảnh truyền thống kết hợp các phương pháp trích xuất đặc trưng thủ công như Histogram of Oriented Gradients (HOG) [5] nhằm cải thiện khả năng biểu diễn của dữ liệu. Tuy nhiên, trong phạm vi dự án, các phương pháp trích xuất đặc trưng bên ngoài không được sử dụng, nhằm giữ cho quy trình xử lý dữ liệu nhất quán giữa các mô hình, đồng thời đảm bảo rằng sự khác biệt về hiệu năng chủ yếu xuất phát từ kiến trúc và khả năng học biểu diễn của từng mô hình.

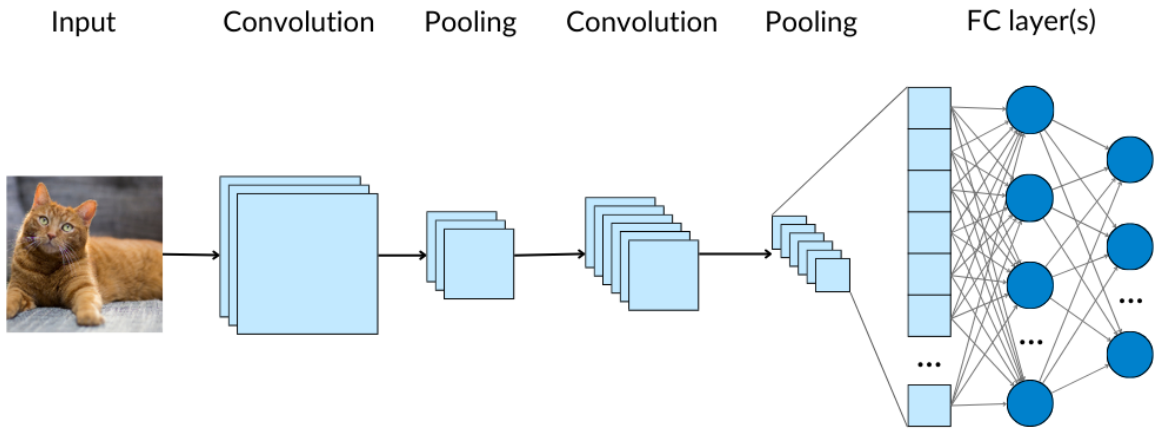
Việc sử dụng ảnh thô ở dạng vector cũng giúp làm rõ những hạn chế của các mô hình học máy cơ bản khi xử lý dữ liệu hình ảnh. Trải phẳng ảnh thành vector không chỉ làm mất cấu trúc không gian của ảnh mà còn khiến số lượng tham số trở nên rất lớn. Điều này vừa đòi hỏi nhiều tài nguyên tính toán, vừa dẫn đến hiện tượng quá khớp (overfitting), tức mô hình cho kết quả tốt trên dữ liệu huấn luyện nhưng lại hoạt động kém với dữ liệu mới. Qua đó ta thấy rõ hơn vai trò của các kiến trúc học sâu trong việc tự động học các đặc trưng không gian hiệu quả.

3.2 Mạng neural tích chập

3.2.1 Tổng quan

Mạng neural tích chập (Convolutional Neural Network – CNN) là một lớp mô hình học sâu được thiết kế chuyên biệt cho các bài toán xử lý dữ liệu hình ảnh [6]. Khác với FCNN hoạt động trên các vector phẳng, CNN tận dụng trực tiếp cấu trúc hai chiều của dữ liệu ảnh. Ý tưởng chính của CNN là sử dụng các phép tích chập (convolution) để phát hiện các đặc trưng không gian, sau đó kết hợp các đặc trưng này qua nhiều tầng để học được các đặc trưng phức tạp hơn.

Một mạng CNN cơ bản bao gồm ba loại tầng chính: tầng tích chập, tầng pooling, và FCNN (Hình 5). Tầng tích chập sử dụng các bộ lọc trượt trên ảnh đầu vào để trích xuất các đặc trưng cục bộ như cạnh, góc hoặc các hoa văn. Nhờ cơ chế chia sẻ trọng số, CNN có số lượng tham số ít hơn đáng kể so với mạng neural truyền thống, đồng thời giảm nguy cơ quá khớp. Các đặc trưng được trích xuất sau đó thường đi qua tầng pooling nhằm giảm kích thước feature map, giảm chi phí tính toán và làm nổi bật đặc trưng quan trọng.



Hình 5: Sơ đồ kiến trúc mạng neural tích chập

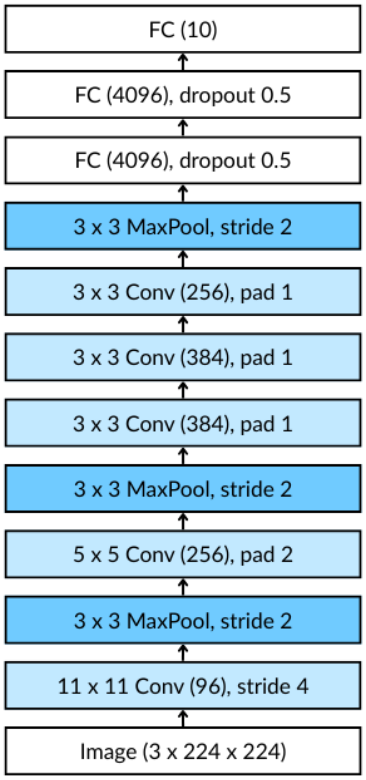
Thông qua việc xếp chồng nhiều lớp tích chập, CNN có khả năng học các biểu diễn phân cấp, trong đó các lớp đầu học đặc trưng mức thấp, còn các lớp sâu hơn học các đặc trưng trừu tượng và mang tính ngữ nghĩa cao hơn. Các lớp tích chập thường được kết hợp với một hoặc nhiều lớp fully-connected ở giai đoạn cuối để thực hiện nhiệm vụ phân loại.

Nhìn chung, CNN là nền tảng của các mô hình phân loại ảnh, cũng như nhiều tác vụ thị giác khác. Các kiến trúc CNN liên tục được cải tiến theo hướng sâu hơn, rộng hơn và hiệu quả hơn, dẫn đến nhiều mô hình hiện đại, trong đó có AlexNet và ResNet.

3.2.2 AlexNet

AlexNet [2] là một trong những kiến trúc mạng neural tích chập đầu tiên thành công trong bài toán phân loại ảnh quy mô lớn. Kiến trúc AlexNet được minh họa trong Hình 6, gồm 8 tầng: 5 tầng convolution và 3 tầng fully-connected. So với các mô hình CNN trước đó, AlexNet có độ sâu lớn hơn và số lượng tham số đáng kể, cho phép mô hình học được các đặc trưng phức tạp hơn từ dữ liệu hình ảnh.

Một số cải tiến quan trọng được giới thiệu trong AlexNet bao gồm việc sử dụng hàm kích hoạt ReLU nhằm tăng tốc độ hội tụ và kỹ thuật dropout để giảm hiện tượng quá khớp. Mô hình cũng phù hợp để huấn luyện song song trên nhiều GPU, giúp tận dụng phần cứng và tăng tốc độ huấn luyện.



Hình 6: Sơ đồ kiến trúc AlexNet

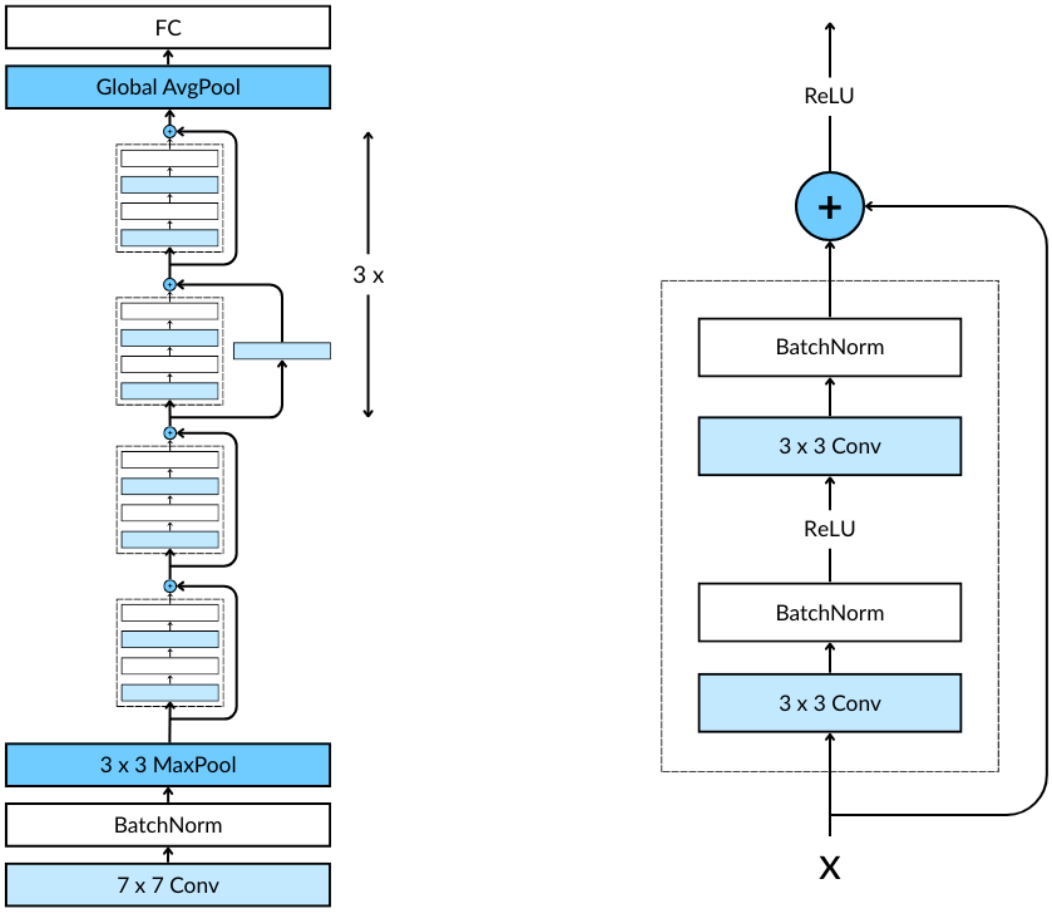
3.2.3 ResNet

ResNet (Residual Network) [3] là một kiến trúc CNN sâu được đề xuất nhằm giải quyết vấn đề suy giảm hiệu năng khi tăng độ sâu của mạng. Trong các mạng CNN truyền thống, việc xếp chồng nhiều lớp tích chập thường dẫn đến hiện tượng suy giảm gradient, khiến quá trình huấn luyện trở nên khó khăn và hiệu năng mô hình không được

cải thiện, thậm chí suy giảm khi mạng trở nên quá sâu.

Để khắc phục vấn đề này, ResNet giới thiệu khái niệm kết nối tắt (skip connection). Thay vì học trực tiếp ánh xạ mong muốn $H(x)$ từ đầu vào x, mỗi khối residual trong ResNet học một hàm $F(x) = H(x) - x$, và đầu ra của khối được tính bằng $y = F(x) + x$. Cơ chế này cho phép gradient được lan truyền trực tiếp qua các kết nối tắt trong quá trình huấn luyện, giúp giảm hiện tượng suy giảm gradient và cho phép huấn luyện các mạng neural có độ sâu lớn hơn đáng kể.

Kiến trúc ResNet được xây dựng từ các khối residual xếp chồng (Hình 7a), trong đó mỗi khối bao gồm một hoặc nhiều lớp tích chập kết hợp với các hàm kích hoạt phi tuyến (Hình 7b). Nhờ thiết kế này, ResNet có khả năng học các biểu diễn sâu và giàu thông tin hơn, đồng thời duy trì sự ổn định trong quá trình huấn luyện.



(a) ResNet (b) Khối residual

Hình 7: Sơ đồ kiến trúc ResNet và khối residual

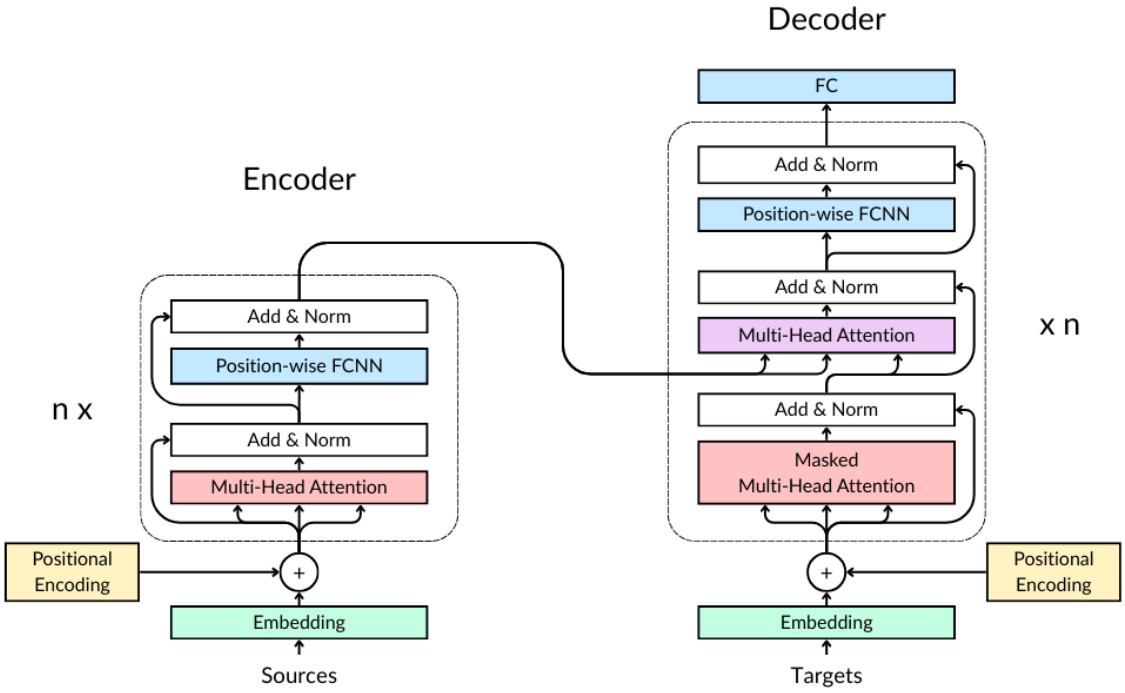
3.3 Vision Transformer

3.3.1 Transformer

Transformer [7] được giới thiệu trong lĩnh vực xử lý dữ liệu chuỗi nhằm giải quyết hạn chế của các mô hình tuần tự như Recurrent Neural Network. Cốt lõi của Transformer là cơ chế self-attention, cho phép mô hình hóa trực tiếp mối quan hệ giữa các phần tử trong chuỗi đầu vào mà không cần xử lý tuần tự. Nhờ đó, mô hình học được các quan hệ dài hạn và cải thiện khả năng song song hóa.

Thành phần cốt lõi của Transformer là cơ chế self-attention, cho phép mỗi phần tử trong chuỗi đầu vào tính toán mức độ liên quan với các phần tử còn lại. Thông qua self-attention, mô hình có khả năng học được các mối quan hệ dài hạn và toàn cục trong dữ liệu, điều mà các kiến trúc dựa trên tích chập hoặc xử lý tuần tự gặp nhiều hạn chế. Trong thực tế, cơ chế multi-head attention thường được sử dụng nhằm cho phép mô hình học các kiểu quan hệ khác nhau trong nhiều không gian biểu diễn song song.

Kiến trúc Transformer được minh họa trong Hình 8 gồm hai thành phần chính: encoder và decoder. Mỗi thành phần bao gồm nhiều lớp self-attention và FCNN, cùng với các kết nối tắt (tương tự ResNet) và chuẩn hóa lớp (Layer Normalization) nhằm đảm bảo ổn định cho quá trình huấn luyện. Encoder nhận chuỗi đầu vào và trích xuất đặc trưng, trong khi decoder dùng thông tin từ encoder để sinh đầu ra. Vì Transformer không có cấu trúc tuần tự, Transformer thiếu thông tin về thứ tự và vị trí của các phần tử. Để khắc phục điều này, thông tin vị trí được bổ sung thông qua các vector positional encoding, giúp mô hình phân biệt được vai trò và vị trí tương đối của từng phần tử trong chuỗi.

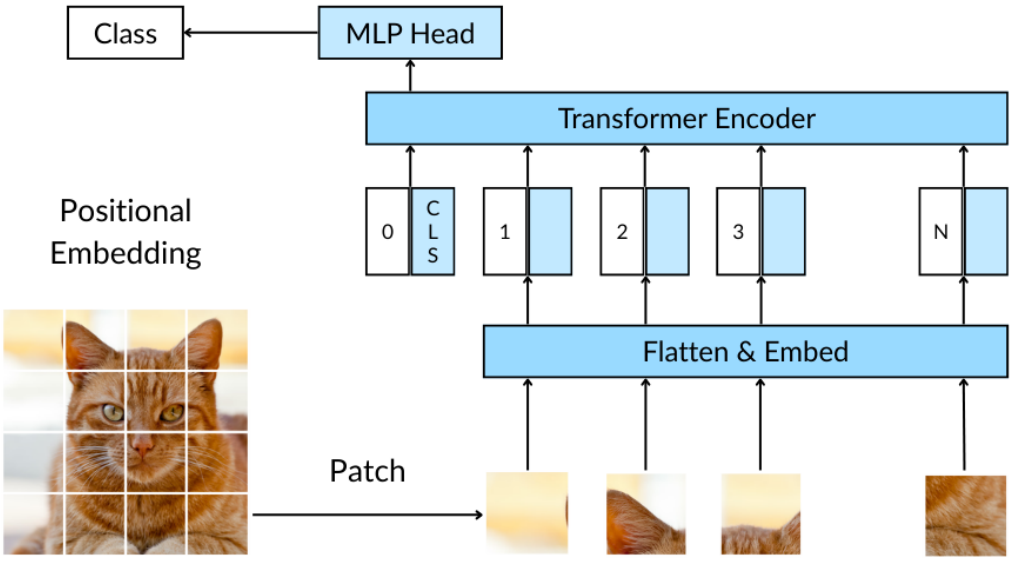


Hình 8: Sơ đồ kiến trúc Transformer

3.3.2 Vision Transformer

Vision Transformer (ViT) [4] là một kiến trúc học sâu áp dụng trực tiếp mô hình Transformer cho bài toán phân loại ảnh. Khác với các mạng CNN truyền thống khai thác cấu trúc không gian cục bộ thông qua phép tích chập, ViT coi ảnh như một chuỗi các phần tử đầu vào và học các mối quan hệ toàn cục giữa các vùng ảnh bằng cơ chế self-attention.

Cơ chế hoạt động của ViT được thể hiện trong Hình 9. Theo đó, ảnh đầu vào được chia thành các mảnh nhỏ (patch) có kích thước cố định, sau đó mỗi patch được trải phẳng và ánh xạ sang một vector đặc trưng thông qua một lớp tuyến tính. Các vector này, cùng với một vector phân loại đại diện cho toàn ảnh, được xem như một chuỗi đầu vào cho Transformer encoder. Thông tin vị trí của các patch được bổ sung thông qua positional encoding nhằm giữ lại cấu trúc không gian của ảnh. Sau khi đi qua toàn bộ kiến trúc, vector phân loại được đưa vào một đầu ra tuyến tính để dự đoán nhãn của ảnh.



Hình 9: Sơ đồ hoạt động của Vision Transformer

Nhờ cơ chế self-attention, ViT có khả năng mô hình hóa mối quan hệ toàn cục giữa các vùng khác nhau trong ảnh, thay vì chỉ khai thác thông tin cục bộ như CNN. Bên cạnh đó, hiệu năng của ViT cải thiện ổn định khi tăng kích thước mô hình hoặc tăng dữ liệu huấn luyện. Do đó, ViT rất được ưa chuộng đối với các tập dữ liệu quy mô lớn.

Tuy nhiên, ViT thường yêu cầu lượng dữ liệu huấn luyện lớn để đạt hiệu năng tốt. Trong điều kiện dữ liệu hạn chế, ViT có thể kém hiệu quả hơn so với các kiến trúc CNN được thiết kế chuyên biệt cho ảnh. Ngoài ra, cơ chế self-attention có độ phức tạp tính toán theo bậc hai đối với số lượng patch, khiến chi phí tăng nhanh khi kích thước ảnh lớn.

4 Triển khai

4.1 Tiền xử lý dữ liệu

Bộ dữ liệu CIFAR-10 được chia thành ba tập con: tập huấn luyện (training set) gồm 40,000 ảnh, tập xác thực (validation set) gồm 10,000 ảnh và tập kiểm thử (test set) gồm 10,000 ảnh. Tập xác thực cho phép đánh giá mô hình trong quá trình huấn luyện và hỗ trợ lựa chọn siêu tham số, trong khi tập kiểm thử giúp đánh giá khả năng tổng quát hóa của mô hình sau khi hoàn tất huấn luyện.

Dữ liệu hình ảnh được tiền xử lý trước hết bằng cách chuyển đổi sang dạng tensor, là cấu trúc dữ liệu dạng mảng nhiều chiều được sử dụng phổ biến trong các thư viện học sâu. Cụ thể, mỗi ảnh màu được biểu diễn dưới dạng tensor ba chiều với các chiều tương ứng là kênh màu (RGB), chiều cao và chiều rộng ảnh.

Sau đó, dữ liệu được chuẩn hóa dựa trên giá trị trung bình và độ lệch chuẩn của tập

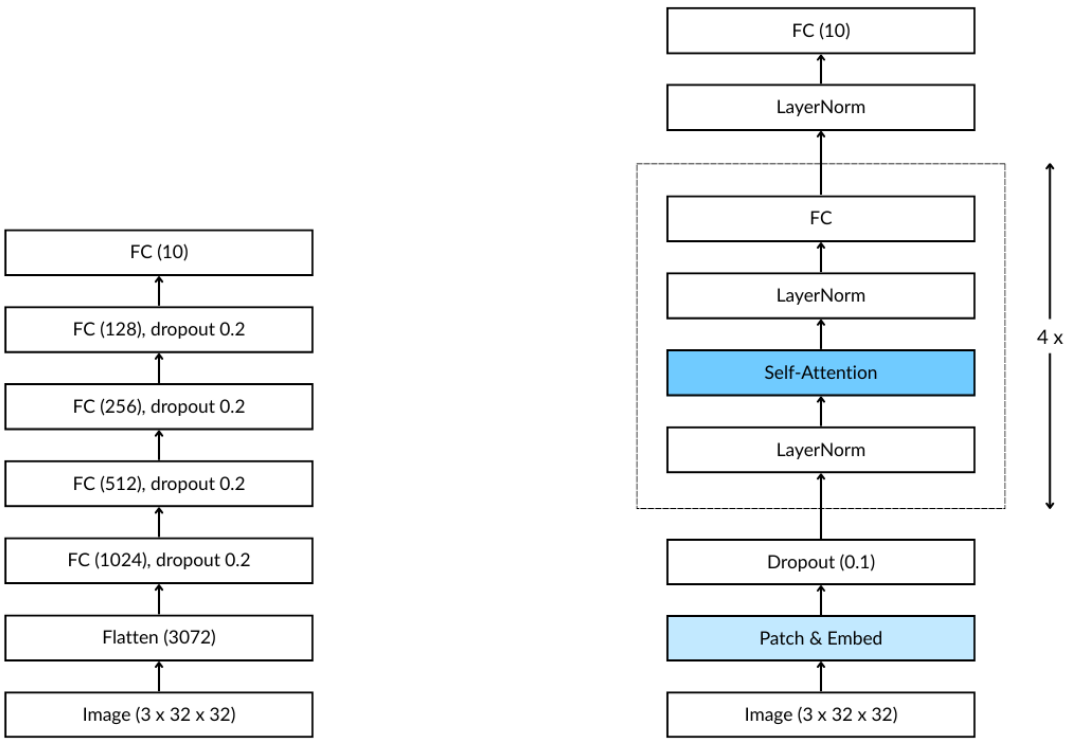
huấn luyện, nhằm đưa phân bố giá trị pixel về cùng một thang đo, giúp quá trình huấn luyện ổn định và hội tụ nhanh hơn. Việc sử dụng số liệu tính từ tập huấn luyện giúp tránh hiện tượng rò rỉ thông tin từ tập xác thực hoặc tập kiểm thử.

Đối với các mô hình yêu cầu kích thước ảnh đầu vào lớn hơn như AlexNet, ResNet và Vision Transformer, kích thước ảnh được thay đổi lên 224×224 pixel. Việc chuẩn hóa và thay đổi kích thước này giúp đảm bảo tính tương thích giữa dữ liệu đầu vào và mô hình, giúp so sánh công bằng hiệu quả giữa các phương pháp khác nhau.

Các bài toán phân loại ảnh thường áp dụng các kỹ thuật tăng cường dữ liệu (data augmentation) nhằm tăng tính đa dạng của tập huấn luyện, qua đó cải thiện khả năng tổng quát hóa và giảm hiện tượng quá khớp. Các phép biến đổi phổ biến bao gồm xoay, lật, thay đổi tỷ lệ hoặc dịch chuyển ảnh. Tuy nhiên trong dự án, các kỹ thuật data augmentation không được sử dụng, nhằm đảm bảo quy trình xử lý dữ liệu nhất quán giữa các mô hình. Việc này giúp đảm bảo rằng sự khác biệt về hiệu năng giữa các mô hình chủ yếu xuất phát từ kiến trúc và khả năng học biểu diễn của mô hình, thay vì từ các kỹ thuật tăng cường dữ liệu bên ngoài.

4.2 Triển khai

Thí nghiệm đã được triển khai trong một Jupyter Notebook, sử dụng thư viện PyTorch và GPU T4 của Colab. Seed khởi tạo được cố định để cho phép so sánh khách quan và đảm bảo khả năng tái lập. Các mô hình được sử dụng bao gồm Softmax Regression, mạng neural truyền thẳng (FCNN), AlexNet, ResNet và Vision Transformer (ViT). Kiến trúc mô hình Softmax Regression, AlexNet và ResNet được trình bày trong Chương **??**. Hình 10 thể hiện kiến trúc FCNN và ViT được sử dụng trong thí nghiệm. Số tham số của mỗi mô hình được trình bày trong Bảng 1.



(a) FCNN (b) ViT

Hình 10: Sơ đồ kiến trúc FCNN và ViT được dùng trong thí nghiệm

Bảng 1: Số tham số của các mô hình

Mô hình	**Số tham số**
Softmax Regression	30,730
FCNN	3,837,066
AlexNet	57,044,810
ResNet	11,184,650
ViT	3,409,674

Softmax Regression và FCNN được huấn luyện trong 15 epoch, các mô hình còn lại được huấn luyện trong 10 epoch do khả năng hội tụ nhanh hơn. Thuật toán tối ưu được sử dụng là Adam với tốc độ học (learning rate) ban đầu là 5×10^{-4}. Trong quá trình huấn luyện, lịch sử huấn luyện (giá trị hàm mất mát trên tập huấn luyện và xác thực) và thời gian mỗi epoch được lưu lại để phục vụ việc vẽ đồ thị và phân tích.

4.3 Tiêu chí đánh giá

Chỉ số đánh giá chính được sử dụng là độ chính xác (accuracy) trên tập kiểm thử. Độ chính xác phản ánh tỷ lệ mẫu được phân loại đúng trên tổng số mẫu và là thước đo phổ biến trong các bài toán phân loại đa lớp với tập dữ liệu cân bằng như CIFAR-10.

Việc lựa chọn accuracy làm chỉ số đánh giá chính giúp kết quả trực quan, dễ diễn giải và thuận tiện cho việc so sánh hiệu năng giữa các mô hình học máy và học sâu khác nhau.

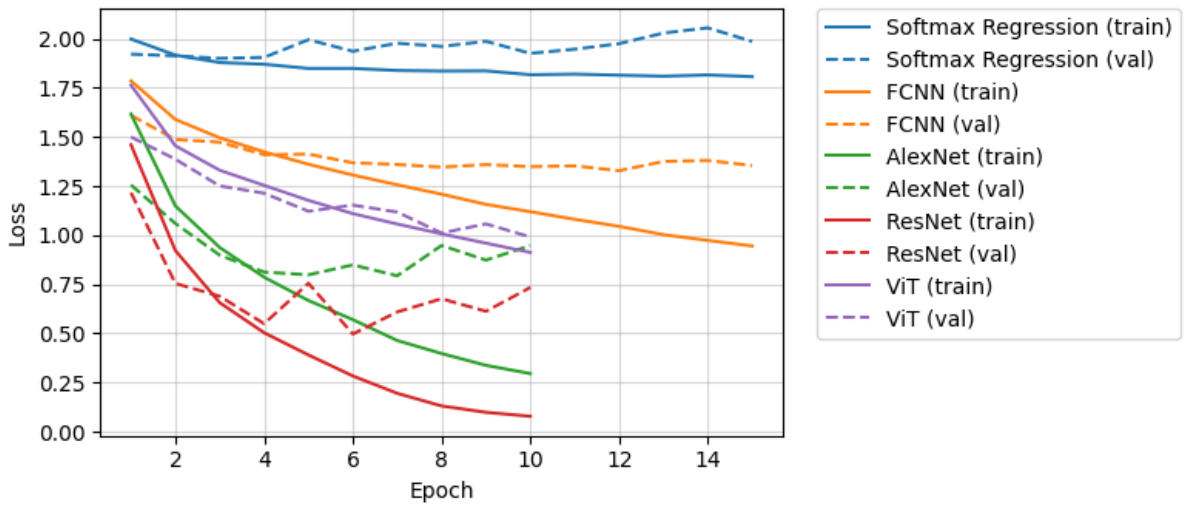
Bên cạnh độ chính xác cuối cùng, các biểu đồ tiến trình huấn luyện theo epoch, bao gồm độ chính xác và hàm mất mát (loss) trên tập huấn luyện và tập xác thực, cũng được sử dụng để phân tích hành vi của mô hình trong quá trình học. Các biểu đồ này cho phép quan sát tốc độ hội tụ, mức độ ổn định của quá trình huấn luyện và giúp phát hiện hiện tượng overfitting.

Ngoài ra, dự án cũng so sánh thời gian trung bình để huấn luyện một epoch giữa các mô hình. Việc đánh giá tốc độ huấn luyện giúp làm rõ sự đánh đổi giữa hiệu năng phân loại và chi phí tính toán, từ đó cung cấp cái nhìn toàn diện hơn về khả năng áp dụng thực tế của từng phương pháp.

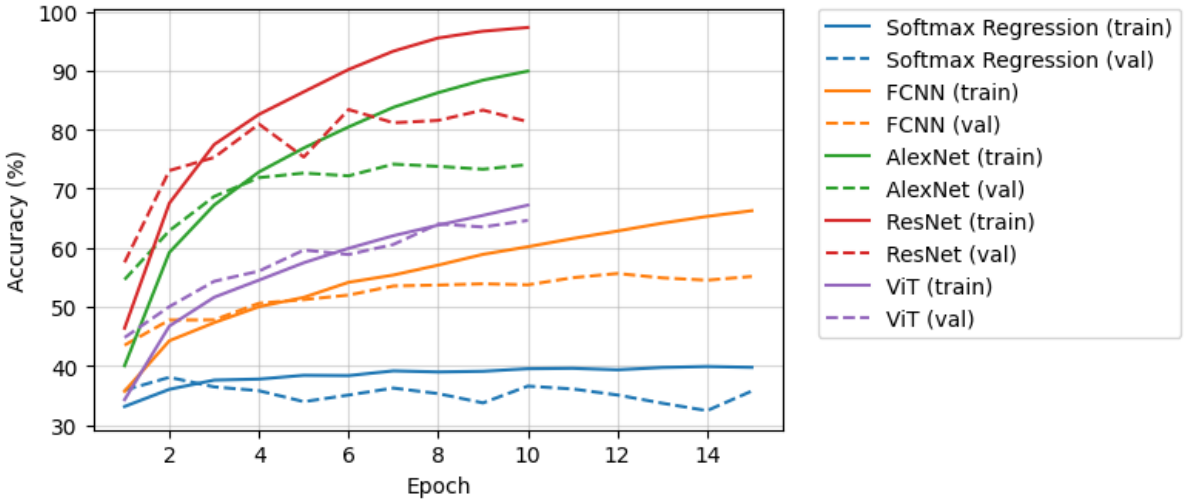
5 Kết quả thực nghiệm

Hình 11 và Hình 12 minh họa sự thay đổi của hàm mất mát và độ chính xác của các mô hình trong suốt quá trình huấn luyện trên tập huấn luyện và tập xác thực. Có thể quan sát thấy rằng mô hình Softmax Regression cho kết quả kém nhất, với giá trị hàm mất mát luôn ở mức cao và độ chính xác thấp trên cả hai tập dữ liệu. Đặc biệt, các đường cong học của mô hình này gần như không có sự cải thiện đáng kể theo số epoch, cho thấy khả năng biểu diễn hạn chế và không phù hợp với bài toán phân loại hình ảnh có độ phức tạp cao. Trong khi đó, mô hình FCNN thể hiện sự cải thiện rõ rệt hơn khi hàm mất mát giảm dần và độ chính xác tăng lên theo thời gian huấn luyện. Tuy nhiên, sự chênh lệch độ chính xác giữa tập huấn luyện và tập xác thực của FCNN lên đến khoảng 10%, phản ánh hiện tượng mô hình học chưa đủ tốt các đặc trưng tổng quát của dữ liệu và còn tồn tại nguy cơ quá khớp.

So với các phương pháp học máy truyền thống, các mô hình CNN cho thấy hiệu năng vượt trội cả về tốc độ hội tụ lẫn độ chính xác đạt được, khi nhanh chóng giảm hàm mất mát trong các epoch đầu và đạt độ chính xác cao trên tập huấn luyện. Tuy nhiên, ở giai đoạn cuối của quá trình huấn luyện, các mô hình CNN xuất hiện dấu hiệu quá khớp, thể hiện qua việc hàm mất mát trên tập xác thực tăng nhẹ và độ chính xác xác thực có xu hướng chững lại. Mặc dù vậy, ResNet vẫn là mô hình đạt kết quả tốt nhất tổng thể, với độ chính xác lần lượt là 97.28% trên tập huấn luyện và 81.33% trên tập xác thực tại epoch cuối cùng. Đối với ViT, mô hình này cho thấy quá trình cải thiện ổn định và nhất quán qua từng epoch, đồng thời độ chính xác giữa hai tập dữ liệu khá tương đồng, cho thấy khả năng tổng quát hóa tốt hơn, dù kết quả cuối cùng vẫn chưa vượt qua các mô hình CNN.



Hình 11: Giá trị hàm mất mát của mỗi mô hình qua các epoch. Đường nét liền là kết quả trên tập huấn luyện, đường nét đứt là kết quả trên tập xác thực.



Hình 12: Độ chính xác của mỗi mô hình qua các epoch

Bảng 2 trình bày độ chính xác trên tập kiểm thử cùng với thời gian huấn luyện trung bình cho mỗi epoch của các mô hình. Kết quả cho thấy ResNet đạt độ chính xác cao nhất trên tập kiểm thử với 81.16%, tiếp theo là AlexNet (73.76%) và ViT (64.55%). Tuy nhiên, hiệu năng cao của ResNet đi kèm với chi phí tính toán lớn, thể hiện qua thời gian huấn luyện trung bình mỗi epoch lên đến 134.11s, cao nhất trong số các mô hình được khảo sát. Ngược lại, Softmax Regression và FCNN có thời gian huấn luyện rất ngắn (khoảng 10–11s mỗi epoch), nhưng độ chính xác thấp hơn đáng kể, cho thấy sự đánh đổi rõ ràng giữa hiệu năng và chi phí tính toán.

Bảng 2: Kết quả thí nghiệm

Mô hình	**Độ chính xác (%)**	**Thời gian epoch trung bình (s)**
Softmax Regression	36.09	**10.26**
FCNN	54.87	11.41
AlexNet	73.76	73.67
ResNet	**81.16**	134.11
ViT	64.55	90.86

Nhìn chung, các kết quả thực nghiệm cho thấy rằng các mô hình học sâu, đặc biệt là CNN, có ưu thế rõ rệt trong bài toán phân loại hình ảnh so với các phương pháp học máy truyền thống. ResNet nổi bật về độ chính xác nhưng yêu cầu tài nguyên tính toán lớn, trong khi ViT thể hiện tiềm năng tốt về khả năng tổng quát hóa dù chưa đạt hiệu suất tối ưu trong thiết lập thí nghiệm này. Kết quả này cho thấy rằng trong điều kiện dữ liệu nhỏ và huấn luyện từ đầu, các kiến trúc CNN vẫn chiếm ưu thế so với Transformer, vốn phát huy hiệu quả tốt hơn với dữ liệu lớn.

6 Kết luận và Hướng phát triển

Dự án đã triển khai và đánh giá các mô hình học máy và học sâu trong bài toán phân loại hình ảnh với bộ dữ liệu CIFAR-10. Nhìn chung, các mô hình có khả năng biểu diễn mạnh hơn, đặc biệt là ResNet, có xu hướng đạt độ chính xác cao hơn so với mô hình tuyến tính hoặc mạng nhỏ, nhưng đi kèm chi phí tính toán lớn hơn. Khoảng cách độ chính xác giữa tập huấn luyện và tập xác thực, kiểm thử cho thấy có hiện tượng overfitting. Để hạn chế hiện tượng này, có thể áp dụng thêm các phương pháp tăng cường dữ liệu (data augmentation) và regularization.

Đối với các mô hình AlexNet, ResNet và ViT, kích thước ảnh đầu vào được phóng đại nhằm tôn trọng yêu cầu kiến trúc gốc, nhưng cũng làm tăng chi phí tính toán và thay đổi thông tin ảnh, ảnh hưởng đến hiệu quả học của mô hình. Có thể xem xét việc điều chỉnh kiến trúc để các mô hình có thể làm việc trực tiếp với kích thước ảnh ban đầu. Ngoài ra, chi phí phần cứng hạn chế có thể giới hạn khả năng thử nghiệm trên các cấu hình lớn hơn. Bên cạnh đó, dữ liệu CIFAR-10 có kích thước nhỏ khiến một số kiến trúc lớn không biểu hiện hết ưu điểm của mình, do đó các dự án tương lai có thể mở rộng thí nghiệm sang các bộ dữ liệu có độ phân giải và độ đa dạng cao hơn để có đánh giá toàn diện.

Tài liệu

[1] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, 2009. Technical Report.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS) 25*. Curran Associates, Inc., 2012.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[5] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[6] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS) 30*, pages 5998–6008, 2017.