ĐẠI HỌC BÁCH KHOA HÀ NỘI
Trường Công nghệ thông tin và Truyền thông

BÁO CÁO DỰ ÁN

Phân loại hình ảnh sử dụng các phương pháp học máy và học sâu

ĐÀM PHÚ ĐẠT
Dat.DP235029@sis.hust.edu.vn

Học phần:	Project 1
Mã học phần:	IT3150
Giảng viên hướng dẫn:	Trần Thị Lan Anh

HÀ NỘI, 11/2025

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

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 vô số ứng dụng, 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 trở thành công cụ chủ đạo trong bài toán phân loại ảnh, mở ra nhiều hướng tiếp cận khác nhau, với những ưu điểm và hạn chế riêng.

Trong bối cảnh đó, em thực hiện dự án này nhằm mục tiêu tìm hiểu sâu cách thức hoạt động của các mô hình từ đơn giản đến hiện đại, đồng thời đánh giá khả năng ứng dụng của chúng trên bộ dữ liệu tiêu 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 đều là đại diện cho các thế hệ mô hình từ học máy truyền thống đến học sâu hiện đại. Thông qua việc huấn luyện và đánh giá các mô hình, em mong muốn hiểu rõ sự khác biệt về tốc độ huấn luyện, độ chính xác, cũng như hành vi cụ thể của từng mô hình. Kết quả so sánh sẽ giúp rút ra nhận xét về ưu nhược điểm của từng kỹ thuật, từ đó định hướng cho việc lựa chọn mô hình trong các bài toán phân loại ảnh tương lai. Mã nguồn đầy đủ và tài liệu bổ sung cho dự án này có sẵn trực tuyến.[1]

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

Chương 2 trình bày về cơ sở lý thuyết, bao gồm tổng quan về bài toán phân loại ảnh và mô tả các mô hình được sử dụng.

Chương 3 mô tả chi tiết quy trình thực nghiệm, bao gồm chuẩn bị dữ liệu, thiết lập mô hình, thông số huấn luyện và các tiêu chí đánh giá

Chương 4 đưa ra kết quả thu được cùng phân tích, nhận xét và so sánh.

Chương 5 tổng hợp các điểm chính, rút ra bài học, và đề xuất các hướng mở cho nghiên cứu tiếp theo.

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

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

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. Một 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...

Bài toán phân loại ảnh có thể được mô tả dưới dạng một hàm ánh xạ $f : \mathbb{R}^{H\times W\times C} \rightarrow \{1, 2, ..., K\}$, trong đó ảnh đầu vào là tensor ba chiều và mô hình cần dự đoán một trong K lớp đầu ra.

2.2 Các phương pháp học máy

2.2.1 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:

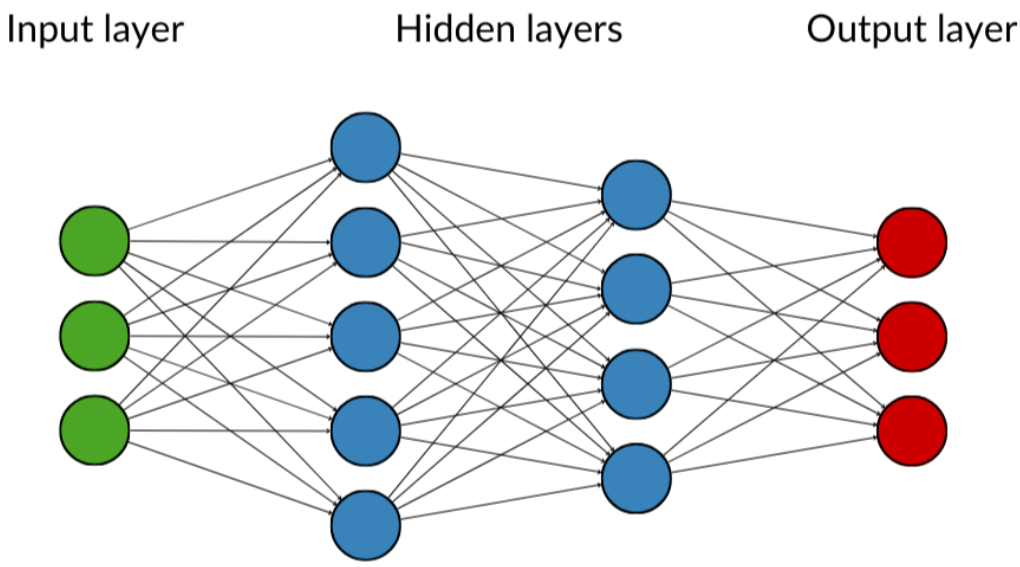
$$\mathcal{L} = -\sum_i \log p(y^{(i)} \mid \mathbf{x}^{(i)})$$

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. Bên cạnh đó, ta cần trải phẳng ảnh thành vector, làm mất cấu trúc không gian của ảnh.

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.

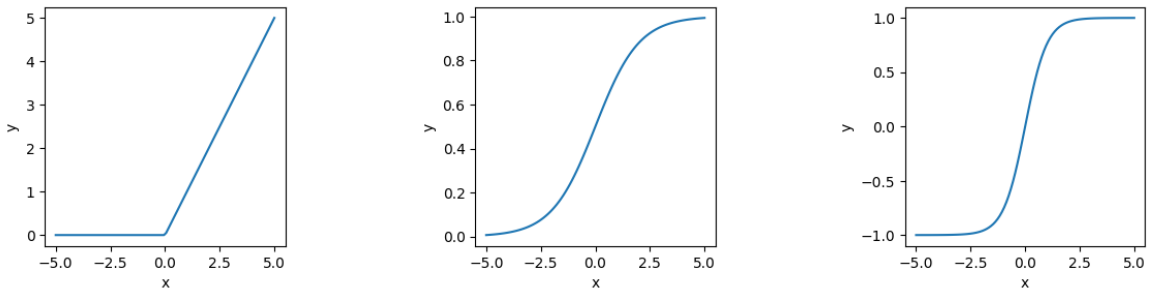
2.2.2 Mạng neural truyền thẳng

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 2.1).



Hình 2.1: 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 2.2), 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 2.2: 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.

Hạn chế của FCNN đối với dữ liệu ảnh là việc 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.

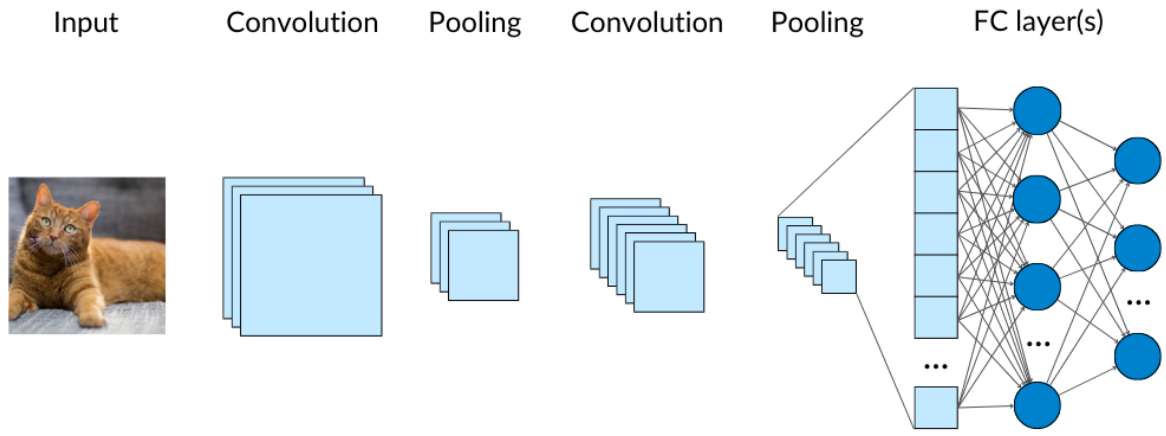
Do đó, trong thị giác máy tính, FCNN thường chỉ được sử dụng như lớp phân loại phía cuối của các mô hình trích xuất đặc trưng như mạng neural tích chập hoặc Transformer. Tuy vậy, FCNN vẫn là mô hình quan trọng làm nền tảng cho các kiến trúc phức tạp hơn.

2.3 Mạng neural tích chập

2.3.1 Tổng quan

Mạng neural tích chập (Convolutional Neural Network – CNN) là kiến trúc cốt lõi trong thị giác máy tính [5]. 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. Các tầng đầu tiên học các đặc trưng cơ bản như cạnh, góc hoặc họa tiết, trong khi các tầng sâu hơn nắm bắt hình dạng, cấu trúc và đối tượng hoàn chỉnh.

Một mạng CNN cơ bản bao gồm ba loại tầng chính: convolution, pooling, và FCNN (Hình 2.3). Ở tầng convolution, các kernel nhỏ được trượt trên ảnh đầu vào để tạo ra các feature map . Nhờ việc chia sẻ trọng số, số lượng tham số của mô hình được giảm đáng kể so với FCNN. 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.

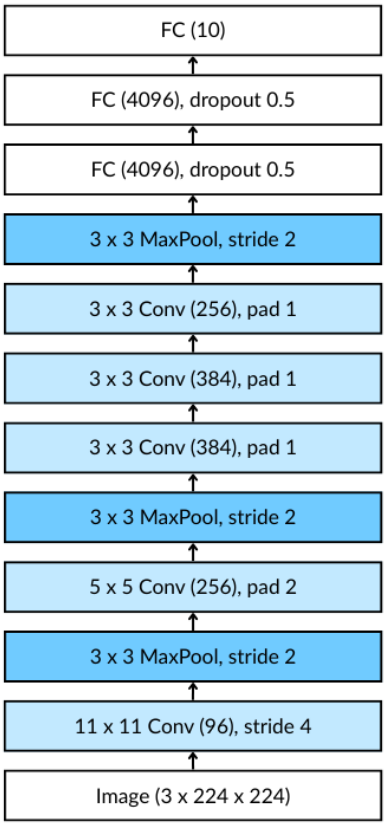


Hình 2.3: Sơ đồ kiến trúc mạng neural tích chập

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.

2.3.2 AlexNet

Kiến trúc AlexNet [2] được minh họa trong Hình 2.4, gồm 8 tầng: 5 tầng convolution và 3 tầng fully-connected.



Hình 2.4: Sơ đồ kiến trúc AlexNet

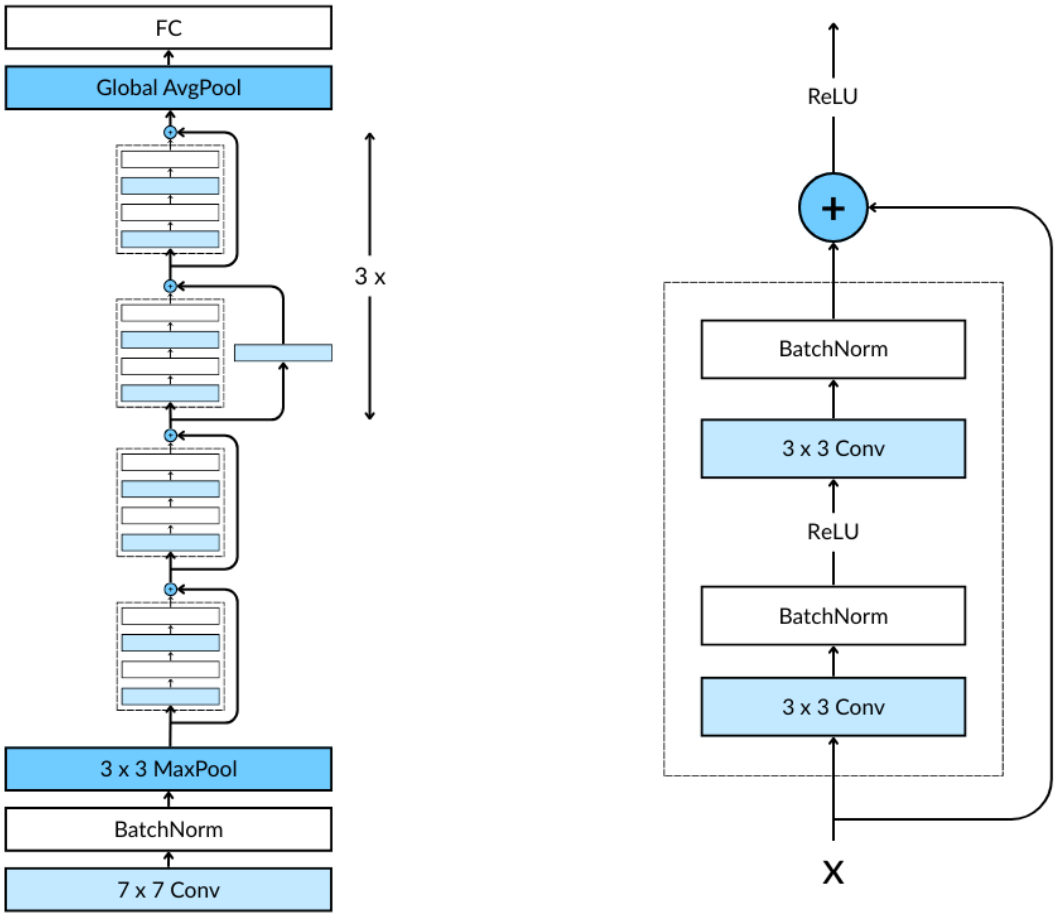
Một điểm nổi bật của AlexNet là việc sử dụng kernel lớn ở các tầng đầu nhằm nhanh chóng trích xuất đặc trưng từ ảnh. Một kỹ thuật quan trọng khác góp phần vào thành công của AlexNet là dropout trong các lớp fully-connected, giúp giảm hiện tượng overfitting - một vấn đề phổ biến khi mô hình có số lượng tham số lớn. Mô hình cũng phù hợp để huấn luyện song song trên nhiều GPU, giúp tận dụng phần cứng và tăng tốc độ huấn luyện.

2.3.3 ResNet

ResNet [3] giải quyết vấn đề suy giảm độ chính xác khi mạng trở nên quá sâu, một hạn chế lớn từng cản trở việc mở rộng mô hình CNN. Trước ResNet, việc tăng số lượng tầng không phải lúc nào cũng cải thiện mô hình. Thậm chí, mạng sâu đôi khi lại hoạt

động kém hơn mạng nông. Hiện tượng này không đơn thuần xuất phát từ overfitting mà liên quan đến khó khăn trong tối ưu hóa, đặc biệt khi gradient phải được truyền qua nhiều tầng.

Kiến trúc ResNet được mô tả trong Hình 2.5a, cấu thành từ các residual block (Hình 2.5b). Ý tưởng cốt lõi của mỗi residual block là kết nối tắt, cho phép tín hiệu đầu vào của một khối đi thẳng đến đầu ra, song song với các tầng học tham số. Nhờ cơ chế này, nếu các tầng học tham số không học được gì hữu ích, mô hình vẫn có thể sao chép đầu vào qua kết nối tắt, giúp quá trình lan truyền gradient ổn định hơn. Nhờ đó mà ResNet có thể huấn luyện các mạng cực kỳ sâu, đồng thời nâng cao đáng kể hiệu suất.



(a) ResNet **(b)** Residual block

Hình 2.5: Sơ đồ kiến trúc ResNet và residual block

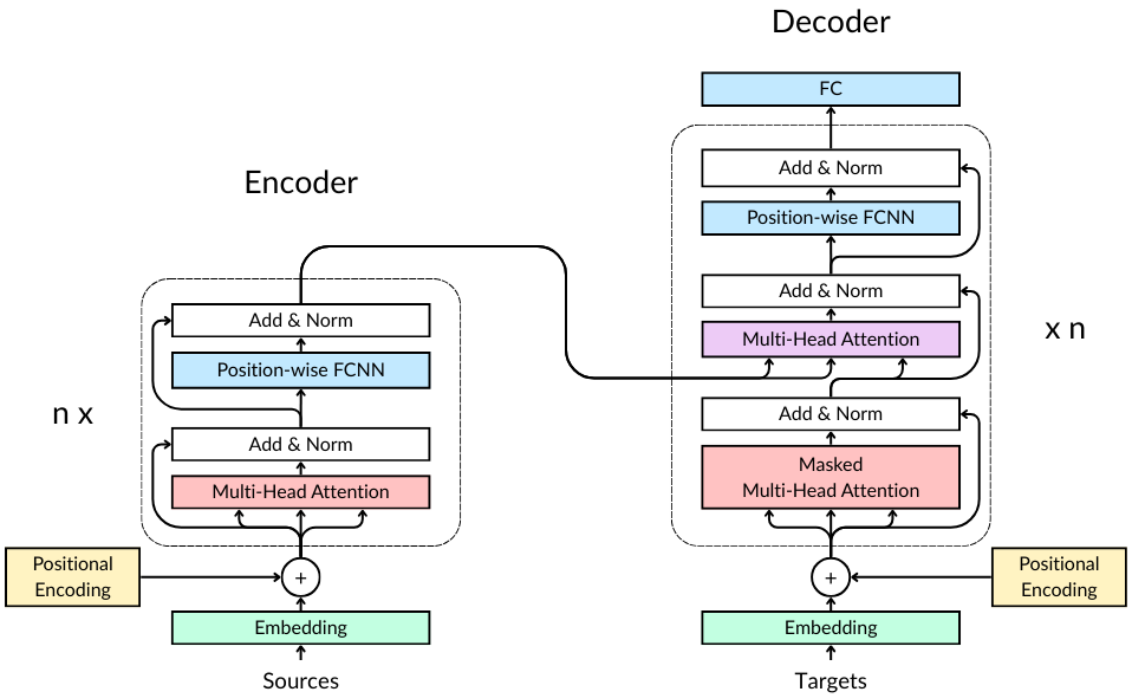
2.4 Vision Transformer

2.4.1 Transformer

Transformer [6] 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 đánh giá mối liên hệ giữa mọi phần tử trong chuỗi mà

không cần xử lý tuần tự. Nhờ đó, mô hình học được các quan hệ dài hạn và có thể huấn luyện song song.

Kiến trúc Transformer được minh họa trong Hình 2.6 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 như RNN, nên vị trí được mã hóa cùng dữ liệu để cung cấp thông tin về trật tự của chuỗi.



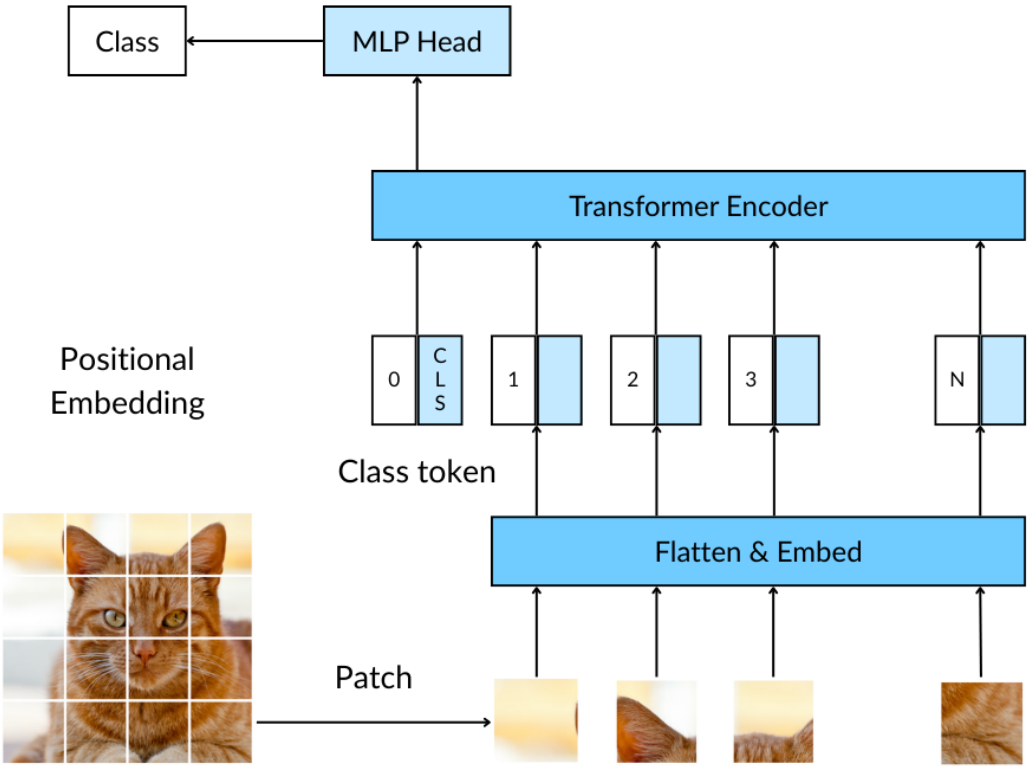
Hình 2.6: Sơ đồ kiến trúc Transformer

2.4.2 Vision Transformer

Vision Transformer (ViT) [4] chứng minh rằng Transformer có thể mở rộng sang lĩnh vực thị giác máy tính, dựa trên ý tưởng rằng hình ảnh có thể được biểu diễn thành một chuỗi các patch tương tự như chuỗi từ trong câu. Nhờ đó, ViT không cần đến các phép tích chập như CNN, nhưng vẫn có khả năng trích xuất đặc trưng từ dữ liệu hình ảnh.

Kiến trúc của ViT được thể hiện trong Hình 2.7. Đầu tiên, ViT chia ảnh đầu vào thành các patch có kích thước cố định, sau đó trải phẳng và biểu diễn mỗi patch thành

một vector. Toàn bộ các vector này tạo thành một chuỗi. Trước khi đi qua Transformer, chuỗi này còn được bổ sung một token đặc biệt – token phân loại – vốn sẽ mang thông tin tổng hợp của toàn bộ ảnh sau khi mô hình xử lý. Sau khi đi qua toàn bộ kiến trúc, token phân loại được đưa vào một đầu ra tuyến tính để dự đoán nhãn của ảnh.



Hình 2.7: Sơ đồ hoạt động của Vision Transformer

Nhờ cơ chế self-attention, ViT có thể học được cấu trúc tổng thể của ảnh, đặc biệt trong những bài toán có nhiều tương quan giữa các vùng ảnh nằm xa nhau. 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.

Ngược lại, với những bộ dữ liệu nhỏ, ViT có thể gặp khó khăn và dễ bị overfitting. 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.

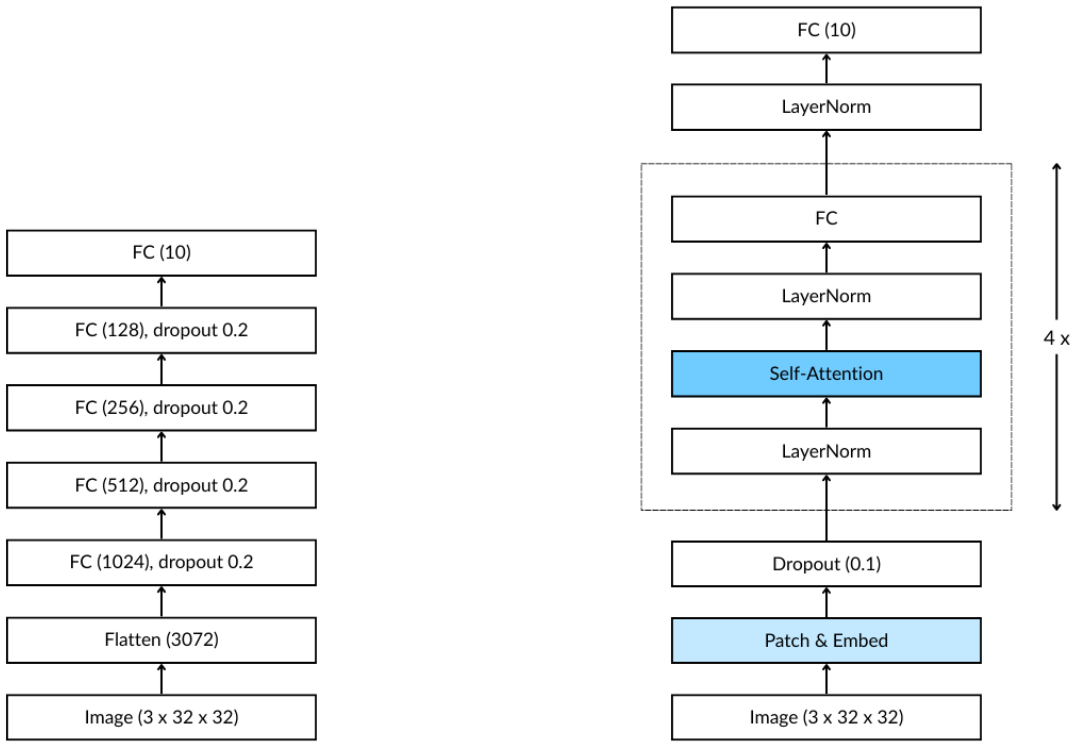
CHƯƠNG 3. TRIỂN KHAI

3.1 Dữ liệu và tiền xử lý

Bộ dữ liệu được sử dụng là CIFAR-10 [1], gồm 60,000 ảnh màu kích thước 32×32 thuộc 10 lớp cần phân loại (Hình 3.x). Tập dữ liệu được chia thành tập huấn luyện gồm 40,000 ảnh, tập xác thực (validation) gồm 10,000 ảnh và tập kiểm thử (test) gồm 10,000 ảnh. Đối với những mô hình lớn như AlexNet, ResNet, ViT yêu cầu ảnh đầu vào có kích thước lớn hơn (224x224), ảnh sẽ được thay đổi về kích thước tương ứng. Ảnh sau đó sẽ được chuyển đổi sang tensor và chuẩn hóa giá trị.

3.2 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 2. Hình 3.1 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 3.1.



(a) FCNN **(b)** ViT

Hình 3.1: Sơ đồ kiến trúc FCNN và ViT được dùng trong thí nghiệm

Bảng 3.1: 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

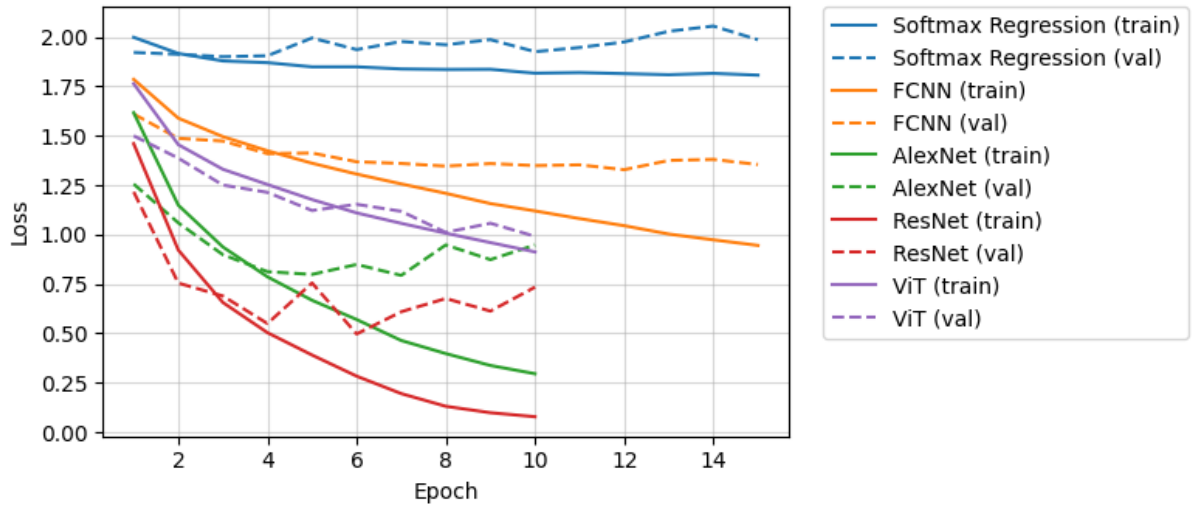
Softmax Regression và FCNN được huấn luyện trong 15 epoch, và các mô hình còn lại được huấn luyện trong 10 epoch vì khả năng hội tụ nhanh hơn. Thuật toán tối ưu hóa được sử dụng là Adam với tốc độ học (learning rate) ban đầu là 5e-4. Trong quá trình huấn luyện, lịch sử huấn luyện (giá trị hàm mất mát 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.

3.3 Tiêu chí đánh giá

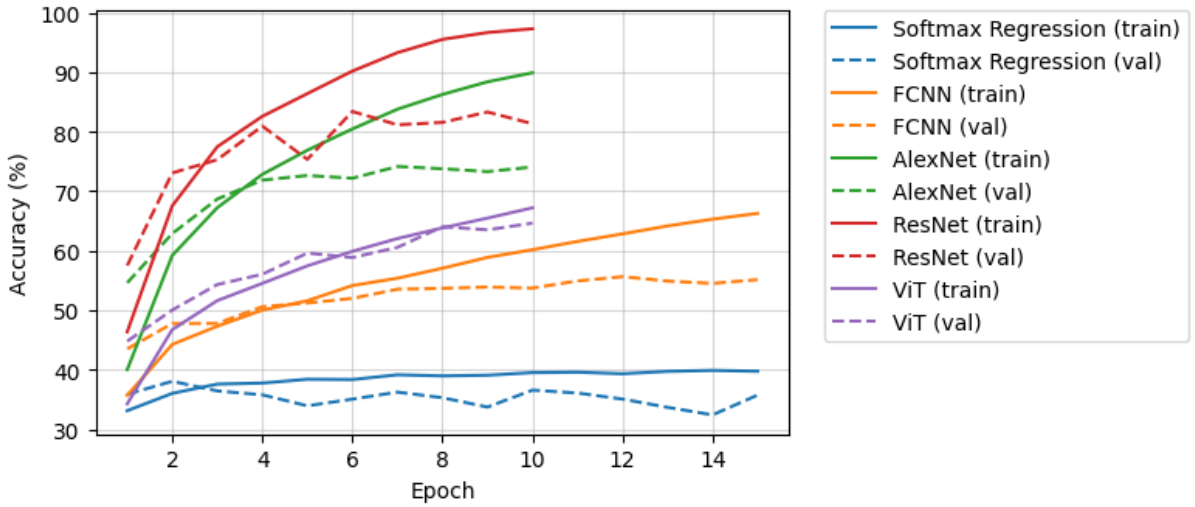
Chỉ số đánh giá chính là độ chính xác (accuracy) trên tập kiểm thử, cùng với các biểu đồ tiến trình huấn luyện để quan sát quá trình hội tụ và phát hiện hiện tượng overfitting hoặc underfitting. Bên cạnh đó, thời gian trung bình để huấn luyện một epoch cũng được so sánh giữa các mô hình để phản ánh chi phí tính toán.

CHƯƠNG 4. KẾT QUẢ THỰC NGHIỆM

Hình 4.1 và Hình 4.2 cho biết giá trị hàm mất mát và độ chính xác của mỗi mô hình trong quá trình huấn luyện. Theo đó, Softmax Regression có độ chính xác thấp và hàm mất mát cao nhất, ít cải thiện qua các epoch. FCNN có sự cải thiện đáng kể, mặc dù độ chính xác trên hai tập dữ liệu chênh lệch đến 10%. Các mô hình CNN có kết quả vượt trội so với các mô hình học máy, nhưng có dấu hiệu quá khớp khi hàm mất mát xác thực tăng nhẹ và độ chính xác xác thực chững lại về cuối quá trình huấn luyện. Tuy vậy, ResNet vẫn có kết quả tốt nhất trên cả tập huấn luyện và tập xác thực (độ chính xác đạt lần lượt 97.28% và 81.33% ở epoch cuối cùng). ViT cải thiện ổn định qua từng epoch, với kết quả trên hai tập dữ liệu rất gần nhau, dù chưa vượt qua các mô hình CNN.



Hình 4.1: 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 4.2: Độ chính xác của mỗi mô hình qua các epoch

Bảng 4.1 cho biết độ chính xác trên tập kiểm thử và thời gian trung bình cho một epoch của các mô hình. Theo đó, ResNet đạt được độ chính xác cao nhất 81.16%, theo sau là AlexNet và ViT. Tuy nhiên, ResNet cũng là mô hình có thời gian huấn luyện một epoch dài nhất, trung bình 134.11s. Các phương pháp học máy Softmax Regression và FCNN có thời gian epoch trung bình ngắn nhất, nhưng độ chính xác thấp hơn đáng kể so với các mô hình hiện đại hơn.

Bảng 4.1: 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

CHƯƠNG 5. 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. 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 THAM KHẢO

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

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

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[4] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2021.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[6] A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems (NIPS) 30*, 2017, pp. 5998–6008.