

KHOA KỸ THUẬT VÀ CÔNG NGHỆ  
BỘ MÔN CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN CHUYÊN NGÀNH  
HỌC KỲ I, NĂM HỌC 2024 – 2025  
**NHẬN DẠNG ĐỐI TƯỢNG  
DỰA TRÊN MÔ HÌNH CNN**

*Giáo viên hướng dẫn:*  
ThS. Nguyễn Mộng Hiền

*Sinh viên thực hiện:*  
Họ tên: Sĩ Chí Thiện  
MSSV: 110121107  
Lớp: DA21TTA

Trà Vinh, tháng 01 năm 2025

KHOA KỸ THUẬT VÀ CÔNG NGHỆ  
BỘ MÔN CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN CHUYÊN NGÀNH  
HỌC KỲ I, NĂM HỌC 2024 – 2025  
**NHẬN DẠNG ĐỐI TƯỢNG  
DỰA TRÊN MÔ HÌNH CNN**

*Giáo viên hướng dẫn:*  
ThS. Nguyễn Mộng Hiền

*Sinh viên thực hiện:*  
Họ tên: Sĩ Chí Thiện  
MSSV: 110121107  
Lớp: DA21TTA

Trà Vinh, tháng 01 năm 2025

*Trà Vinh, ngày 05 tháng 01 năm 2025*

---

1

## NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

*Trà Vinh, ngày 05 tháng 01 năm 2025*

**Thành viên hội đồng**  
(Ký tên và ghi rõ họ tên)

### LỜI CẢM ƠN

Trước hết, tôi xin bày tỏ lòng biết ơn sâu sắc đến thầy Nguyễn Mộng Hiền, giảng viên hướng dẫn của tôi. Trong suốt quá trình thực hiện đồ án "Nhận dạng đối tượng dựa trên mô hình CNN," thầy đã tận tình hướng dẫn và truyền đạt những kiến thức chuyên sâu, giúp tôi hiểu rõ hơn về các nguyên lý của mạng nơ-ron tích chập và ứng dụng của chúng trong việc nhận dạng đối tượng. Sự hỗ trợ tận tâm và các góp ý giá trị của thầy đã giúp tôi không chỉ hoàn thiện đồ án mà còn phát triển tư duy nghiên cứu và kỹ năng giải quyết vấn đề.

Tôi cũng xin gửi lời cảm ơn chân thành đến Khoa Kỹ thuật và Công nghệ, Trường Đại học Trà Vinh, nơi đã cung cấp môi trường học tập và nghiên cứu thuận lợi. Những tài liệu, giáo trình và trang thiết bị mà Khoa đã cung cấp là nguồn tri thức và công cụ quý giá, giúp tôi có nền tảng kiến thức vững chắc để tiếp cận các khía cạnh phức tạp của đề tài.

Dù đã nỗ lực hết mình, tôi hiểu rằng đồ án vẫn còn những hạn chế do tính phức tạp của đề tài và kinh nghiệm của bản thân. Tôi rất mong nhận được những góp ý quý báu từ quý thầy cô để tiếp tục hoàn thiện kiến thức và kỹ năng trong tương lai.

Tôi xin chân thành cảm ơn!

## MỤC LỤC

<b>LỜI CẢM ƠN .....</b>	<b>3</b>
<b>MỤC LỤC .....</b>	<b>4</b>
<b>DANH MỤC HÌNH.....</b>	<b>6</b>
<b>DANH MỤC TỪ VIẾT TẮT .....</b>	<b>7</b>
<b>MỞ ĐẦU.....</b>	<b>9</b>
<b>CHƯƠNG 1: TỔNG QUAN .....</b>	<b>12</b>
<b>CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT .....</b>	<b>13</b>
2.1. Tổng quan về Mạng Nơ-ron Tích chập (CNN).....	13
2.1.1. Khái quát về Mạng Nơ-ron Tích chập .....	13
2.1.2. Cơ chế hoạt động của mô hình CNN .....	14
2.1.3. Ứng dụng thực tiễn của CNN trong thị giác máy tính.....	14
2.2. Các thành phần chính trong CNN .....	15
2.2.1 Lớp tích chập (Convolutional Layer) .....	15
2.2.2 Lớp gộp (Pooling Layer).....	15
2.2.3 Lớp kết nối đầy đủ (Fully Connected Layer).....	16
2.3. Các kỹ thuật cải thiện hiệu suất của CNN .....	17
2.3.1. Regularization trong mô hình CNN.....	17
2.3.2. Kỹ thuật Dropout.....	18
2.3.3. Batch Normalization.....	18
2.4. Tiền xử lý dữ liệu trong mô hình CNN.....	19
2.4.1. Chuẩn hóa dữ liệu và thay đổi kích thước ảnh.....	19
2.4.2. Kỹ thuật tăng cường dữ liệu (Data Augmentation) .....	20
2.5. Tổng quan về tập dữ liệu CIFAR-10 .....	21
2.5.1. Đặc điểm và cấu trúc của tập dữ liệu.....	21
2.5.2. Lý do lựa chọn tập dữ liệu cho nghiên cứu .....	22
2.6. Công cụ và thư viện hỗ trợ.....	23
2.6.1. Python và các thư viện TensorFlow và Pytorch .....	23
2.6.2. Nền tảng Google Colab và các tiện ích tích hợp .....	23
2.7. Đánh giá hiệu suất mô hình .....	24
2.7.1. Các chỉ số đánh giá hiệu suất chính trong phân loại hình ảnh .....	24

2.7.2. Phương pháp phân tích kết quả huấn luyện .....	25
2.8. So sánh CNN với các mô hình học sâu khác .....	26
<b>CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU .....</b>	<b>27</b>
3.1. Mô tả bài toán .....	27
3.2. Tiền xử lý dữ liệu .....	28
3.2.1. Tải và phân tích dữ liệu CIFAR-10 với TensorFlow .....	28
3.2.2. Trực quan hóa dữ liệu bằng Seaborn .....	28
3.2.3. Chuẩn hóa dữ liệu và mã hóa nhãn bằng One-Hot Encoding .....	29
3.2.4. Tăng cường dữ liệu bằng ImageDataGenerato.....	29
3.3. Xây dựng kiến trúc mô hình CNN .....	30
3.3.1. Thiết kế kiến trúc CNN.....	30
3.3.2. Triển khai mô hình CNN.....	35
3.4. Đánh giá hiệu suất mô hình .....	36
<b>CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU .....</b>	<b>38</b>
4.1. Kết quả huấn luyện.....	38
4.2 Dự đoán trên tập kiểm tra .....	40
4.2.1 Kết quả dự đoán chính xác.....	40
4.2.2 Kết quả dự đoán sai .....	40
4.2.3 Dự đoán hình ảnh thực tế.....	41
4.3 Hiệu suất tổng thể.....	43
<b>CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>45</b>
5.1. Kết luận.....	45
5.2. Hạn chế.....	45
5.3. Phương hướng phát triển .....	45
<b>DANH MỤC TÀI LIỆU THAM KHẢO .....</b>	<b>46</b>
<b>PHỤ LỤC .....</b>	<b>47</b>

## DANH MỤC HÌNH

Hình 2.1 Minh họa cấu luồng xử lý của mô hình CNN [1].....	13
Hình 2.4 Minh họa quá trình tích chập và tạo bản đồ đặc trưng [2] .....	15
Hình 2.5 Minh họa quá trình gộp tối đa trên bản đồ đặc trưng [3] .....	16
Hình 2.6 Minh họa hình ảnh lớp Fully Connected Layer [2] .....	16
Hình 2.7 Minh họa hình ảnh đại diện từ 10 lớp của tập dữ liệu CIFAR-10. [5].....	21
Hình 2.8 Minh họa logo của Google Colab.....	24
Hình 3.1 Minh họa ảnh đầu vào (Input) .....	27
Hình 3.2 Minh họa kết quả đầu ra (Output) .....	27
Hình 3.3 Minh họa lưu đồ kiến trúc CNN.....	34
Hình 4.1 Minh họa biểu đồ độ chính xác (Accuracy).....	38
Hình 4.2 Minh họa biểu đồ hàm mất mát (Loss) .....	39
Hình 4.3 Minh họa hình ảnh kết quả sau khi huấn luyện.....	39
Hình 4.4 Minh họa hình ảnh những đoán chính xác.....	40
Hình 4.5 Minh họa hình ảnh những dự đoán sai.....	41
Hình 4.6 Minh họa hình ảnh chức năng tải ảnh lên .....	41
Hình 4.7 Minh họa hình ảnh chọn ảnh từ thư mục .....	42
Hình 4.8 Minh họa hình ảnh đang tiến hành dự đoán.....	42
Hình 4.9 Minh họa hình ảnh kết quả dự đoán .....	43



**DANH MỤC TỪ VIẾT TẮT**

<b>STT</b>	<b>KÝ HIỆU VIẾT TẮT</b>	<b>NỘI DUNG VIẾT TẮT</b>
1	BN	Batch Normalization
2	CNN	Convolutional Neural Network
3	GPU	Graphics Processing Unit
4	ReLU	Rectified Linear Unit
5	TPU	Tensor Processing Unit
6	L2	L2 Regularization
7	RMSprop	Root Mean Square Propagation
8	NLP	Natural Language Processing

## TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

Đề tài "Nhận dạng đối tượng dựa trên mô hình CNN" tập trung vào việc nghiên cứu và ứng dụng Mạng Nơ-ron Tích chập (Convolutional Neural Network - CNN) trong lĩnh vực thị giác máy tính, đặc biệt là trong nhiệm vụ nhận dạng và phân loại đối tượng từ hình ảnh. Mục tiêu chính của đề tài là tìm hiểu sâu về mô hình CNN, bao gồm cả lý thuyết nền tảng và cách thức triển khai thực tế, nhằm xây dựng một hệ thống có khả năng nhận dạng đối tượng trong ảnh với độ chính xác cao.

Trong quá trình thực hiện, đề tài sẽ phân tích cấu trúc chi tiết của CNN, từ các lớp tích chập (convolutional layers), lớp gộp (pooling layers), đến các lớp kết nối đầy đủ (fully connected layers).

Hệ thống sẽ nhận đầu vào là các hình ảnh chứa đối tượng cần nhận dạng, sau đó xử lý qua mô hình CNN để trích xuất các đặc trưng quan trọng và đưa ra dự đoán về nhãn của đối tượng trong ảnh. Quá trình này bao gồm các bước tiền xử lý dữ liệu như thay đổi kích thước ảnh, chuẩn hóa dữ liệu, và tăng cường dữ liệu (data augmentation) để làm phong phú tập dữ liệu huấn luyện.

Về công nghệ, đề tài sử dụng ngôn ngữ lập trình Python kết hợp với các thư viện học sâu mạnh mẽ như TensorFlow và Keras để xây dựng và huấn luyện mô hình CNN. Để tối ưu quá trình triển khai, môi trường Google Colab được lựa chọn nhờ khả năng hỗ trợ tính toán trên GPU miễn phí, giúp tăng tốc đáng kể cho việc huấn luyện mô hình với các tập dữ liệu lớn.

## MỞ ĐẦU

### 1. Lý do chọn đề tài

Trong bối cảnh phát triển mạnh mẽ của trí tuệ nhân tạo, đặc biệt trong các ứng dụng nhận dạng và phân tích hình ảnh. Các hệ thống tự động hóa dựa trên AI, như nhận dạng đối tượng qua hình ảnh, giúp tối ưu hóa quy trình, tăng độ chính xác và hiệu quả trong xử lý dữ liệu. Đặc biệt trong nhận dạng và phân tích hình ảnh, đang ngày càng chứng tỏ vai trò quan trọng trong tự động hóa quy trình. Các hệ thống AI nhận dạng đối tượng giúp tăng độ chính xác, hiệu suất và đáp ứng nhanh trong xử lý hình ảnh, tạo nền tảng ứng dụng rộng rãi trong các lĩnh vực.

Mạng Nơ-ron Tích chập (Convolutional Neural Network - CNN) là một trong những mô hình học sâu hàng đầu, đã chứng minh hiệu quả vượt trội trong nhận dạng đối tượng từ hình ảnh nhờ khả năng trích xuất đặc trưng và phân loại chính xác. CNN giúp xử lý dữ liệu hình ảnh phức tạp và là nền tảng cho các ứng dụng như xe tự hành, nhận dạng khuôn mặt, phân loại y tế, và giám sát an ninh.

Ngoài độ chính xác cao, CNN còn tối ưu hóa quy trình nhận dạng, tiết kiệm thời gian và công sức trong phân tích hình ảnh, tạo nền tảng cho các hệ thống tự động hiệu quả. Ứng dụng CNN không chỉ nâng cao trải nghiệm người dùng mà còn đem lại lợi ích kinh tế, xã hội. Đề tài "Nhận dạng đối tượng dựa trên mô hình CNN" được chọn với mong muốn đóng góp vào sự phát triển của AI và cung cấp các giải pháp thực tiễn cho thời đại số.

### 2. Mục tiêu nghiên cứu

**Nghiên cứu cấu trúc và cơ chế hoạt động của mô hình CNN**, bao gồm các lớp tích chập, lớp kết nối đầy đủ và các thành phần chính khác trong mạng nơ-ron tích chập, nhằm nắm bắt cách thức hoạt động của mô hình trong nhận dạng và phân loại hình ảnh.

**Tìm hiểu và triển khai các bước tiền xử lý dữ liệu** cho tập dữ liệu hình ảnh, bao gồm các kỹ thuật chuẩn hóa và chuyển đổi định dạng ảnh để phù hợp với đầu vào của mô hình CNN, đảm bảo chất lượng và tính nhất quán của dữ liệu trong quá trình huấn luyện.

**Triển khai và huấn luyện mô hình CNN trên nền tảng Google Colab**, sử dụng các thư viện như TensorFlow hoặc PyTorch, để xây dựng mô hình có khả năng phân loại hình ảnh với độ chính xác cao, đồng thời tận dụng các tài nguyên GPU để tối ưu hóa quá trình huấn luyện.

**Đánh giá hiệu quả của mô hình CNN** dựa trên các chỉ số như độ chính xác và độ nhạy, phân tích khả năng nhận dạng đối tượng trên tập dữ liệu và xác định các yếu tố ảnh hưởng đến hiệu suất của mô hình, từ đó đưa ra các hướng cải tiến phù hợp.

### 3. Phương pháp nghiên cứu

#### **Nghiên cứu tài liệu và lý thuyết về CNN:**

- Tìm hiểu lý thuyết nền tảng của Mạng Nơ-ron Tích chập (CNN), bao gồm các lớp tích chập, lớp kết nối đầy đủ và các thành phần khác của mô hình.
- Phân tích cấu trúc và cơ chế hoạt động của CNN để hiểu cách mô hình này xử lý và trích xuất đặc trưng từ hình ảnh trong bài toán nhận dạng.

#### **Tiền xử lý dữ liệu hình ảnh:**

- Tiến hành các bước tiền xử lý cho tập dữ liệu CIFAR-10, bao gồm chuẩn hóa dữ liệu, thay đổi kích thước và chuyển đổi định dạng ảnh phù hợp với yêu cầu đầu vào của mô hình CNN.
- Áp dụng các kỹ thuật tăng cường dữ liệu (data augmentation) để tạo sự đa dạng cho tập dữ liệu, giúp mô hình học tốt hơn từ các biến đổi của đối tượng trong ảnh.

#### **Xây dựng và triển khai mô hình CNN:**

- Sử dụng ngôn ngữ Python và các thư viện học sâu như TensorFlow hoặc PyTorch để thiết kế và xây dựng mô hình CNN.
- Triển khai quá trình huấn luyện mô hình trên nền tảng Google Colab, tận dụng GPU miễn phí để tối ưu hóa thời gian và hiệu quả huấn luyện.

#### **Đánh giá và tối ưu hóa mô hình:**

- Đánh giá hiệu quả của mô hình CNN qua các chỉ số như độ chính xác, độ nhạy, độ đặc hiệu và ma trận nhầm lẫn để phân tích hiệu suất phân loại.

#### **Phân tích kết quả và đề xuất hướng phát triển:**

- Phân tích các yếu tố ảnh hưởng đến hiệu suất của mô hình CNN trong nhận dạng hình ảnh.
- Đưa ra các nhận xét, kết luận và đề xuất hướng cải tiến mô hình, như điều chỉnh kiến trúc hoặc thử nghiệm với các phương pháp tiền xử lý và tối ưu hóa khác, nhằm nâng cao hiệu quả trong các nghiên cứu tiếp theo.

### 4. Đối tượng nghiên cứu

- Kiến trúc Mạng Nơ-ron Tích chập (CNN)

- Tập dữ liệu hình ảnh CIFAR-10
- Nghiên cứu ngôn ngữ lập trình Python và các thư viện TensorFlow, Keras
- Môi trường Google Colab

## **5. Phạm vi nghiên cứu**

Nghiên cứu tập trung vào ứng dụng của Mạng Nơ-ron Tích chập (CNN) trong nhận dạng đối tượng từ ảnh tĩnh, với dữ liệu chính là tập hình ảnh CIFAR-10. Mô hình CNN được triển khai trên nền tảng Google Colab, sử dụng ngôn ngữ Python và các thư viện học sâu như TensorFlow và Keras. Phạm vi nghiên cứu bao gồm quy trình tiền xử lý dữ liệu, huấn luyện và đánh giá hiệu suất của mô hình dựa trên các chỉ số chính như độ chính xác, nhằm tạo ra một mô hình CNN có khả năng nhận dạng đối tượng hiệu quả.

## CHƯƠNG 1: TỔNG QUAN

Trong bối cảnh phát triển mạnh mẽ của trí tuệ nhân tạo và thị giác máy tính, các phương pháp nhận dạng đối tượng dựa trên mô hình Mạng Nơ-ron Tích chập (CNN) ngày càng khẳng định vai trò quan trọng nhờ khả năng phân tích hình ảnh hiệu quả. Nghiên cứu này được thực hiện nhằm giải quyết vấn đề nhận dạng đối tượng trong hình ảnh, ứng dụng CNN để phát hiện và phân loại các đối tượng thuộc các lớp khác nhau. Việc sử dụng mô hình CNN không chỉ giúp tự động hóa các nhiệm vụ phân tích mà còn tăng độ chính xác và hiệu quả trong xử lý dữ liệu hình ảnh.

Để thực hiện nghiên cứu, tập dữ liệu CIFAR-10 với 10 lớp đối tượng đặc trưng đã được lựa chọn. Nghiên cứu được triển khai trên nền tảng Google Colab, sử dụng ngôn ngữ lập trình Python cùng các thư viện TensorFlow và Keras. Quy trình thực hiện bao gồm các bước như tiền xử lý dữ liệu, thiết kế kiến trúc CNN, huấn luyện mô hình và đánh giá hiệu suất. Các kỹ thuật hiện đại như tăng cường dữ liệu, Dropout, và Regularization được áp dụng nhằm tối ưu hóa khả năng học tập và giảm thiểu hiện tượng quá khớp.

Sau quá trình nghiên cứu và thử nghiệm, mô hình CNN được xây dựng đã đạt được kết quả khả quan với độ chính xác cao trên tập dữ liệu kiểm tra, đáp ứng yêu cầu đặt ra. Kết quả này mở ra tiềm năng ứng dụng trong nhiều lĩnh vực như phân loại hình ảnh y tế, giám sát an ninh, và nhận dạng đối tượng trong môi trường công nghiệp, với thời gian xử lý hợp lý và hiệu suất ổn định.

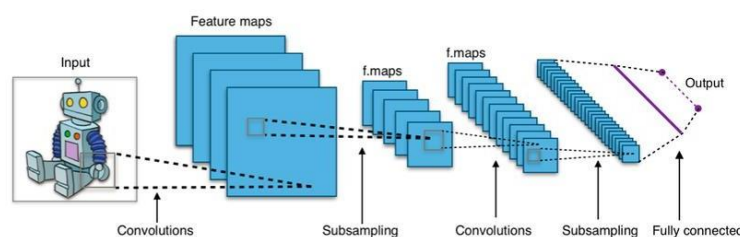
## CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

### 2.1. Tổng quan về Mạng Nơ-ron Tích chập (CNN)

#### 2.1.1. Khái quát về Mạng Nơ-ron Tích chập

Mạng Nơ-ron Tích chập (Convolutional Neural Network - CNN) là một loại mô hình học sâu (Deep Learning) được thiết kế đặc biệt để xử lý và phân tích dữ liệu dạng lưới, chẳng hạn như hình ảnh và chuỗi thời gian. CNN được lấy cảm hứng từ cấu trúc hệ thần kinh thị giác của con người, nơi các tế bào thần kinh phản ứng với các kích thích cụ thể như cạnh hoặc góc của đối tượng. Điểm nổi bật của CNN nằm ở khả năng tự động học và trích xuất các đặc trưng quan trọng từ dữ liệu đầu vào mà không cần thiết kế thủ công. Điều này giúp CNN trở thành một công cụ mạnh mẽ trong nhiều bài toán thị giác máy tính, bao gồm nhận dạng đối tượng, phân loại hình ảnh, nhận diện khuôn mặt và phát hiện vật thể.

CNN hoạt động dựa trên sự kết hợp của ba loại lớp chính. Các lớp tích chập (convolutional layers) chịu trách nhiệm trích xuất đặc trưng từ ảnh đầu vào, tiếp theo là các lớp gộp (pooling layers) giúp giảm kích thước dữ liệu để tăng hiệu quả tính toán. Cuối cùng, các lớp kết nối đầy đủ (fully connected layers) thực hiện việc tổng hợp thông tin và đưa ra quyết định phân loại. Chính nhờ cách thức hoạt động này, CNN vừa hiệu quả trong xử lý dữ liệu phức tạp vừa giữ được tính linh hoạt, đáp ứng tốt các yêu cầu của những bài toán thị giác máy tính hiện đại [1].



Hình 2.1 Minh họa cấu luồng xử lý của mô hình CNN [1]

#### Một số điểm mạnh nổi bật của CNN bao gồm:

- Tự động trích xuất đặc trưng mà không cần thiết kế thủ công.
- Xử lý hiệu quả dữ liệu hình ảnh, giảm tham số nhờ các phép tích chập.
- Mở rộng ứng dụng sang xử lý ngôn ngữ, nhận diện chữ viết và phân tích chuỗi thời gian.

CNN là một nền tảng quan trọng trong lĩnh vực học sâu, nổi bật với khả năng học hiệu quả và tính linh hoạt cao. Mô hình này không ngừng được cải tiến và ứng dụng rộng rãi, mang lại nhiều giá trị thiết thực trong nghiên cứu và công nghiệp.

### 2.1.2. Cơ chế hoạt động của mô hình CNN

Mạng Nơ-ron Tích chập (CNN) được thiết kế với kiến trúc bao gồm nhiều lớp liên kết, giúp xử lý và phân tích dữ liệu hình ảnh một cách hiệu quả. CNN thường được cấu thành từ ba loại lớp chính: lớp tích chập (Convolutional Layer), lớp gộp (Pooling Layer) và lớp kết nối đầy đủ (Fully Connected Layer). Mỗi loại lớp đảm nhiệm một vai trò quan trọng, từ trích xuất đặc trưng đến phân loại đối tượng.

Lớp tích chập là thành phần cơ bản nhất, chịu trách nhiệm trích xuất các đặc trưng cục bộ từ ảnh đầu vào bằng cách sử dụng các bộ lọc (filters) di chuyển trên ảnh để phát hiện các yếu tố như cạnh, góc hoặc các họa tiết phức tạp hơn. Tiếp theo, lớp gộp được sử dụng để giảm kích thước bản đồ đặc trưng, giúp giảm số lượng tham số và tăng hiệu quả tính toán. Cuối cùng, dữ liệu được đưa qua lớp kết nối đầy đủ, nơi các đặc trưng đã trích xuất được tổng hợp để đưa ra dự đoán cuối cùng.

Một mô hình CNN điển hình bao gồm các thành phần chính: ảnh đầu vào (ảnh RGB kích thước 32x32x3), các lớp tích chập để trích xuất đặc trưng, các lớp gộp để giảm kích thước dữ liệu, và lớp kết nối đầy đủ để dự đoán kết quả. Nhờ cách tổ chức này, CNN có thể xử lý dữ liệu hình ảnh hiệu quả và mang lại độ chính xác cao trong các bài toán nhận dạng và phân loại [1].

### 2.1.3. Ứng dụng thực tiễn của CNN trong thị giác máy tính

Mạng Nơ-ron Tích chập (CNN) đã trở thành một công cụ mạnh mẽ trong lĩnh vực thị giác máy tính nhờ khả năng tự động trích xuất và phân tích các đặc trưng từ hình ảnh. CNN được ứng dụng rộng rãi trong nhiều bài toán thực tế, giúp giải quyết các vấn đề phức tạp với độ chính xác cao.

- **Nhận diện khuôn mặt:** CNN được sử dụng trong các hệ thống nhận diện khuôn mặt, từ quản lý truy cập, bảo mật cho đến nhận diện người dùng trên thiết bị di động.
- **Phân tích hình ảnh y tế:** CNN hỗ trợ chẩn đoán và phát hiện bệnh lý từ các ảnh y khoa như X-quang, MRI, và CT scan.
- **Giám sát và phân tích an ninh:** Trong các hệ thống giám sát, CNN giúp nhận diện đối tượng, theo dõi chuyển động và phát hiện hành vi bất thường từ video hoặc hình ảnh, góp phần nâng cao an ninh trong công cộng.
- **Xe tự hành:** CNN là nền tảng trong các hệ thống xe tự hành, giúp nhận diện biển báo giao thông, vật cản, và người đi bộ.



- **Kiểm tra chất lượng sản phẩm:** Trong ngành công nghiệp, CNN được sử dụng để kiểm tra và phát hiện lỗi trên dây chuyền sản xuất, đảm bảo chất lượng sản phẩm.
- **Tìm kiếm và phân loại hình ảnh:** Các nền tảng trực tuyến ứng dụng CNN để gợi ý, tìm kiếm và phân loại hình ảnh theo nội dung, cải thiện trải nghiệm người dùng.

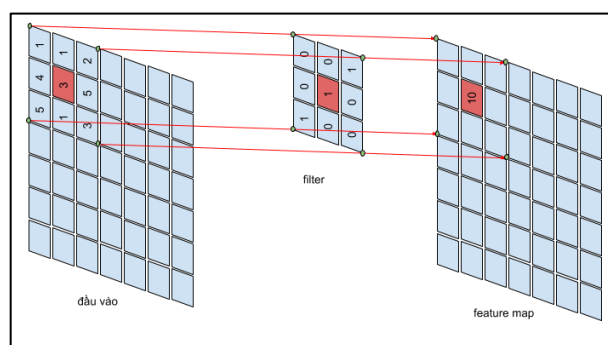
Những ứng dụng này minh chứng cho sự linh hoạt và tiềm năng của CNN trong việc xử lý hình ảnh và dữ liệu phức tạp, mang lại giá trị lớn trong nhiều lĩnh vực, từ y tế, giao thông đến công nghiệp và đời sống hàng ngày.

## 2.2. Các thành phần chính trong CNN

### 2.2.1 Lớp tích chập (Convolutional Layer)

Lớp tích chập (Convolutional Layer) là thành phần chính trong Mạng Nơ-ron Tích chập (CNN), đảm nhiệm vai trò trích xuất các đặc trưng cục bộ từ dữ liệu đầu vào. Quá trình này thực hiện bằng cách sử dụng các bộ lọc (filters) nhỏ như 3x3 hoặc 5x5 quét qua toàn bộ ảnh, giúp phát hiện các đặc trưng như cạnh, góc, hoặc họa tiết phức tạp. Kết quả tạo ra là một bản đồ đặc trưng (feature map) chứa thông tin trích xuất.

Các tham số như kích thước bộ lọc, bước nhảy (stride) và kỹ thuật padding ảnh hưởng đến chất lượng và kích thước bản đồ đặc trưng. Lớp tích chập giúp giảm số lượng tham số và tăng khả năng khái quát hóa, cho phép mô hình học từ các đặc trưng cơ bản đến phức tạp hơn, đồng thời duy trì hiệu quả trong việc xử lý dữ liệu lớn.



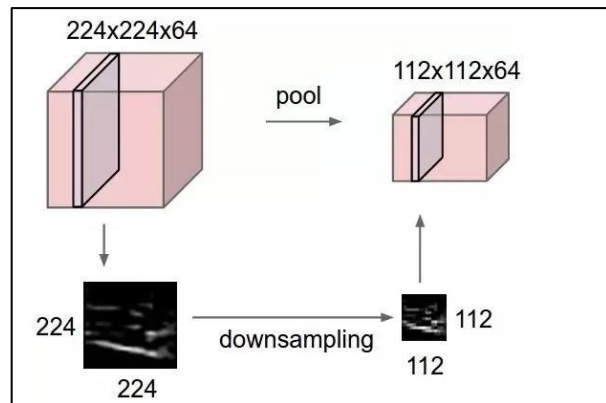
Hình 2.2 Minh họa quá trình tích chập và tạo bản đồ đặc trưng [2]

### 2.2.2 Lớp gộp (Pooling Layer)

Lớp gộp (Pooling Layer) trong CNN đóng vai trò giảm kích thước bản đồ đặc trưng, giúp giảm số lượng tham số và tính toán mà vẫn giữ được các thông tin quan trọng. Điều này tăng khả năng khái quát hóa của mô hình, giúp nó hoạt động hiệu quả hơn trên cả dữ liệu đã biết và dữ liệu mới.

Lớp gộp hoạt động bằng cách áp dụng một cửa sổ trượt (thường là  $2 \times 2$  hoặc  $3 \times 3$ ) trên bản đồ đặc trưng, lấy giá trị đại diện từ mỗi vùng được chọn. Có hai phương pháp gộp phổ biến:

- **Max pooling:** Chọn giá trị lớn nhất trong cửa sổ trượt. Phương pháp này thường được sử dụng vì nó giữ lại các đặc trưng nổi bật nhất.
- **Average pooling:** Tính trung bình các giá trị trong cửa sổ trượt, giúp làm mịn thông tin đặc trưng. [3]

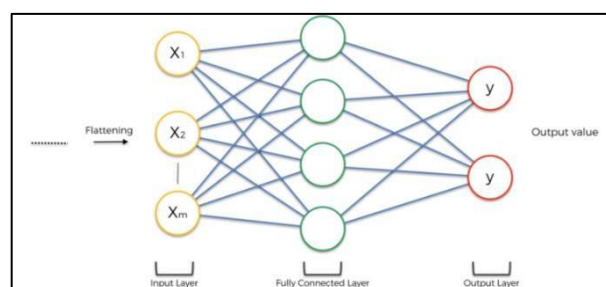


Hình 2.3 Minh họa quá trình gộp tối đa trên bản đồ đặc trưng [3]

### 2.2.3 Lớp kết nối đầy đủ (Fully Connected Layer)

Lớp kết nối đầy đủ (Fully Connected Layer) là phần cuối cùng trong kiến trúc Mạng Nơ-ron Tích chập (CNN), nơi dữ liệu đã được trích xuất đặc trưng qua các lớp trước được sử dụng để đưa ra dự đoán cuối cùng. Lớp này kết nối tất cả các neuron từ lớp trước với các neuron hiện tại, giống như cách hoạt động của mạng nơ-ron truyền thống (Feedforward Neural Network).

Dữ liệu đầu vào cho lớp kết nối đầy đủ thường là một vector một chiều. Trước khi đưa vào lớp này, các bản đồ đặc trưng từ lớp tích chập hoặc lớp gộp sẽ được làm phẳng (flatten). Lớp kết nối đầy đủ sau đó thực hiện các phép tính tuyến tính, kết hợp với các hàm kích hoạt như ReLU hoặc Softmax, để học các mối quan hệ phức tạp giữa các đặc trưng và dự đoán nhãn đầu ra.



Hình 2.4 Minh họa hình ảnh lớp Fully Connected Layer [2]

### Vai trò chính của lớp kết nối đầy đủ:

- **Ra quyết định phân loại:** Lớp này đảm nhận vai trò phân loại dựa trên các đặc trưng đã được trích xuất, thường đưa ra kết quả cuối cùng dưới dạng xác suất cho từng nhãn.
- **Tạo kết nối toàn cục:** Không giống các lớp tích chập và gộp chỉ tập trung vào thông tin cục bộ, lớp kết nối đầy đủ xử lý tất cả các đặc trưng, kết hợp chúng để hiểu tổng thể dữ liệu.

Lớp kết nối đầy đủ đóng vai trò quan trọng trong việc chuyển đổi các đặc trưng đã học thành thông tin trực quan, dễ hiểu và phù hợp với bài toán. Mặc dù có thể yêu cầu nhiều tham số, nhưng khi kết hợp với các lớp trước đó, nó giúp CNN đạt hiệu quả cao trong phân loại và nhận dạng [3].

## 2.3. Các kỹ thuật cải thiện hiệu suất của CNN

### 2.3.1. Regularization trong mô hình CNN

Regularization là một kỹ thuật quan trọng trong học máy, đặc biệt trong Mạng Nơ-ron Tích chập (CNN), nhằm giảm thiểu hiện tượng quá khớp (overfitting). Hiện tượng này xảy ra khi mô hình học quá mức các đặc điểm từ tập dữ liệu huấn luyện, dẫn đến hiệu suất kém trên dữ liệu mới. Regularization giúp cải thiện khả năng tổng quát hóa, đảm bảo mô hình hoạt động hiệu quả trên các tập dữ liệu chưa từng thấy.

Kỹ thuật Regularization được áp dụng thông qua các ràng buộc hoặc sửa đổi trong quá trình huấn luyện. Các phương pháp phổ biến bao gồm:

**L2 Regularization (Ridge Regularization):** Phương pháp này bổ sung một giá trị phạt tỉ lệ thuận với bình phương của trọng số vào hàm mất mát, khuyến khích mô hình duy trì trọng số nhỏ hơn. Điều này giúp giảm nguy cơ mô hình học các đặc trưng nhiễu không cần thiết.

**L1 Regularization (Lasso Regularization):** Tương tự như L2 Regularization, nhưng sử dụng giá trị tuyệt đối của trọng số thay vì bình phương. Điều này dẫn đến việc làm mờ hoặc loại bỏ các trọng số không quan trọng, tạo ra một mô hình đơn giản và dễ khái quát hóa hơn.

**Early Stopping:** Phương pháp này dừng quá trình huấn luyện khi hiệu suất trên tập kiểm tra không còn cải thiện, ngay cả khi lỗi trên tập huấn luyện tiếp tục giảm. Điều này giúp ngăn ngừa mô hình học quá mức từ tập huấn luyện.

**Weight Decay:** Là một dạng L2 Regularization được tích hợp trực tiếp vào thuật toán tối ưu hóa (như SGD hoặc Adam), giúp kiểm soát kích thước trọng số và duy trì sự ổn định trong quá trình huấn luyện.

Những kỹ thuật này không chỉ cải thiện hiệu suất của mô hình mà còn giúp tăng độ tin cậy và khả năng áp dụng của mô hình CNN vào các bài toán thực tế.

### 2.3.2. Kỹ thuật Dropout

Dropout là một kỹ thuật thường được sử dụng trong CNN để giảm thiểu hiện tượng quá khớp (overfitting). Phương pháp này hoạt động bằng cách "tắt" ngẫu nhiên một số neuron trong quá trình huấn luyện, giúp mô hình học được các đặc trưng khái quát hơn thay vì quá phụ thuộc vào một số neuron cụ thể.

#### Cơ chế hoạt động của Dropout:

- Trong quá trình huấn luyện, một tỷ lệ neuron được chọn ngẫu nhiên để tạm thời ngừng hoạt động.
- Tỷ lệ Dropout (thường từ 0.2 đến 0.5) xác định số neuron bị tắt, tỷ lệ 0.5 nghĩa là 50% neuron không hoạt động.
- Khi suy luận, tất cả neuron được kích hoạt và trọng số được điều chỉnh để bù đắp cho Dropout.

#### Lợi ích của Dropout:

- **Ngăn chặn quá khớp:** Bằng cách tắt ngẫu nhiên các neuron, Dropout làm giảm khả năng mô hình học quá chi tiết từ tập dữ liệu huấn luyện.
- **Tăng khả năng tổng quát hóa:** Dropout buộc mô hình phải học các đặc trưng quan trọng hơn, cải thiện hiệu suất trên dữ liệu chưa từng thấy.
- **Tăng tính ngẫu nhiên:** Kỹ thuật này tạo ra nhiều "biến thể nhỏ" của mạng, giúp mô hình học tập phong phú hơn.

Dropout thường được sử dụng ở các lớp kết nối đầy đủ (Fully Connected Layer), nơi chứa nhiều tham số và dễ bị quá khớp. Dropout hiệu quả khi áp dụng đúng tỷ lệ. Nếu tỷ lệ quá cao, mô hình có thể mất thông tin quan trọng.

### 2.3.3. Batch Normalization

Batch Normalization (BN) là một kỹ thuật giúp quá trình huấn luyện mô hình CNN ổn định hơn và nhanh hơn. Bằng cách chuẩn hóa các giá trị đầu ra của mỗi lớp trong mạng nơ-ron, BN đảm bảo rằng dữ liệu không bị biến đổi quá lớn trong quá trình lan truyền, giúp mô hình dễ học hơn.

### **Batch Normalization thực hiện hai bước chính:**

**Chuẩn hóa dữ liệu:** Các giá trị đầu ra từ lớp trước (gọi là "activations") được chuẩn hóa dựa trên trung bình và độ lệch chuẩn trong từng batch dữ liệu. Điều này giúp các giá trị luôn nằm trong khoảng ổn định.

**Tái điều chỉnh:** Sau khi chuẩn hóa, BN cho phép học thêm hai tham số mới ( $\gamma$  và  $\beta$ ) để mô hình tự điều chỉnh nếu cần thiết, nhằm giữ nguyên khả năng học của mạng.

### **Lợi ích của Batch Normalization:**

- **Huấn luyện nhanh hơn:** BN làm giảm hiện tượng "internal covariate shift" (sự thay đổi phân phối dữ liệu qua các lớp), giúp mô hình học hiệu quả hơn ngay cả với learning rate cao.
- **Ổn định hơn:** Dữ liệu được chuẩn hóa sẽ giúp giảm rủi ro mô hình trở nên không ổn định trong quá trình huấn luyện.
- **Giảm quá khớp:** BN có tác dụng giống một dạng Regularization nhẹ, làm cho mô hình hoạt động tốt hơn trên dữ liệu chưa từng thấy.
- **Ít phụ thuộc vào khởi tạo:** Mô hình không quá nhạy cảm với cách chọn trọng số ban đầu hoặc các siêu tham số khác.

BN hoạt động hiệu quả nhất khi kích thước batch đủ lớn, vì nó tính toán trung bình và độ lệch chuẩn trên toàn bộ batch. Đối với batch nhỏ, các giá trị này có thể không chính xác, dẫn đến hiệu suất giảm.

## **2.4. Tiền xử lý dữ liệu trong mô hình CNN**

### **2.4.1. Chuẩn hóa dữ liệu và thay đổi kích thước ảnh**

Chuẩn hóa dữ liệu và thay đổi kích thước ảnh là hai bước quan trọng trong tiền xử lý dữ liệu hình ảnh cho mô hình CNN. Những kỹ thuật này đảm bảo rằng dữ liệu đầu vào phù hợp với yêu cầu của mô hình, từ đó tăng tốc độ huấn luyện đồng thời cải thiện độ chính xác.

### **Chuẩn hóa dữ liệu hình ảnh (Normalization):**

Dữ liệu hình ảnh ban đầu được biểu diễn dưới dạng ma trận các giá trị điểm ảnh, thường nằm trong khoảng từ 0 đến 255. Chuẩn hóa là việc chuyển đổi các giá trị này về một phạm vi cố định, thường là  $[0, 1]$ , hoặc đưa chúng về phân phối chuẩn với trung bình là 0 và phương sai là 1.

Phương pháp phổ biến là chia tất cả giá trị điểm ảnh cho 255, hoặc trừ đi giá trị trung bình và chia cho độ lệch chuẩn của tập dữ liệu.

### **Thay đổi kích thước ảnh (Resizing):**

CNN yêu cầu đầu vào có kích thước cố định (32x32 hoặc 224x224 điểm ảnh) để đảm bảo tương thích với kiến trúc mô hình. Do đó, ảnh từ các nguồn khác nhau cần được thay đổi kích thước trước khi đưa vào mô hình. Việc thay đổi kích thước giúp đảm bảo đồng nhất dữ liệu và giảm độ phức tạp tính toán.

Các công cụ như OpenCV hoặc PIL thường được sử dụng để thay đổi kích thước. Chẳng hạn, một ảnh ban đầu có kích thước 200x300 điểm ảnh có thể được thay đổi về 32x32 điểm ảnh để giảm số lượng tham số và tăng hiệu quả xử lý mà vẫn giữ lại các đặc trưng quan trọng.

Tầm quan trọng của các bước tiền xử lý này là chuẩn hóa và thay đổi kích thước ảnh không chỉ giúp mô hình học nhanh hơn mà còn giảm rủi ro về lỗi kỹ thuật do dữ liệu không đồng nhất. Đồng thời, dữ liệu được chuẩn hóa và đồng nhất kích thước giúp cải thiện độ chính xác và khả năng khái quát hóa của mô hình trên các tập dữ liệu chưa từng thấy.

### **2.4.2. Kỹ thuật tăng cường dữ liệu (Data Augmentation)**

Kỹ thuật tăng cường dữ liệu (Data Augmentation) là một phương pháp quan trọng trong tiền xử lý dữ liệu cho CNN. Bằng cách tạo ra các biến thể của dữ liệu ban đầu, Data Augmentation giúp mở rộng tập dữ liệu huấn luyện, cải thiện khả năng tổng quát hóa của mô hình và giảm nguy cơ quá khớp (overfitting).

#### **Nguyên lý hoạt động:**

Tăng cường dữ liệu dựa trên việc áp dụng các phép biến đổi ngẫu nhiên hoặc cố định lên dữ liệu ban đầu, tạo ra các mẫu mới mà không cần thu thập thêm dữ liệu. Những biến đổi này giúp mô hình học cách nhận diện đối tượng trong các điều kiện đa dạng như góc nhìn khác nhau, ánh sáng thay đổi, hoặc biến dạng hình học.

Các phương pháp tăng cường dữ liệu phổ biến:

- **Lật ảnh (Flipping):** Lật ngang hoặc lật dọc ảnh để mô phỏng các góc nhìn khác nhau của đối tượng.
- **Xoay ảnh (Rotation):** Xoay ảnh một góc bất kỳ để làm phong phú tập dữ liệu.
- **Dịch chuyển (Translation):** Dịch chuyển ảnh theo trục X hoặc Y để tạo các biến thể vị trí của đối tượng.

- **Phóng to và thu nhỏ (Scaling):** Tăng hoặc giảm kích thước đối tượng trong ảnh mà không làm thay đổi bản chất của đối tượng.
- **Biến đổi màu sắc (Color Jitter):** Thay đổi độ sáng, độ tương phản, hoặc màu sắc để mô phỏng các điều kiện ánh sáng khác nhau.
- **Nhiều ngẫu nhiên (Random Noise):** Thêm nhiễu vào ảnh để giúp mô hình trở nên linh hoạt và ít nhạy cảm hơn với dữ liệu nhiễu.

#### Lợi ích của Data Augmentation:

Mở rộng tập dữ liệu: Tăng cường dữ liệu tạo ra một tập dữ liệu đa dạng hơn từ số lượng ảnh ban đầu, đặc biệt hữu ích khi tập dữ liệu nhỏ.

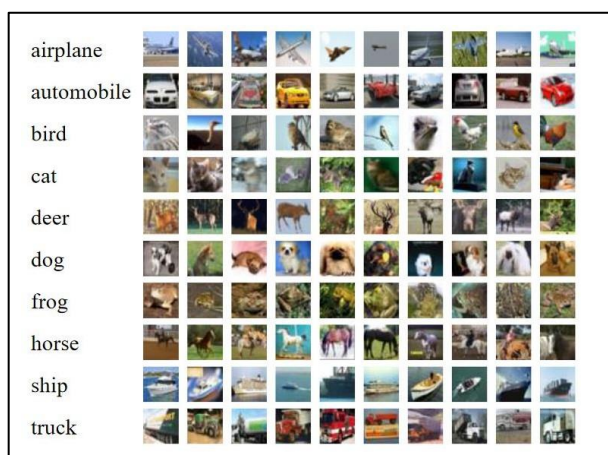
Cải thiện khả năng tổng quát hóa: Mô hình học được cách nhận dạng đối tượng trong nhiều điều kiện khác nhau, giảm hiện tượng quá khớp.

Tăng tính linh hoạt của mô hình: Giúp mô hình hoạt động tốt hơn trên các ảnh không hoàn toàn giống với dữ liệu huấn luyện.

## 2.5. Tổng quan về tập dữ liệu CIFAR-10

### 2.5.1. Đặc điểm và cấu trúc của tập dữ liệu

CIFAR-10 (Canadian Institute For Advanced Research) là một trong những tập dữ liệu tiêu chuẩn được sử dụng rộng rãi trong các bài toán học sâu (Deep Learning) và thị giác máy tính. Tập dữ liệu này được thiết kế để đánh giá khả năng phân loại hình ảnh của các mô hình học máy, đặc biệt là các mạng nơ-ron sâu như CNN.



Hình 2.5 Minh họa hình ảnh đại diện từ 10 lớp của tập dữ liệu CIFAR-10. [5]

#### Đặc điểm chính của CIFAR-10:

- **Số lượng hình ảnh:** Tập dữ liệu gồm tổng cộng 60,000 hình ảnh màu. Trong đó: 50,000 hình ảnh được sử dụng để huấn luyện mô hình, 10,000 hình ảnh được sử dụng để kiểm tra (test) mô hình.

- Kích thước hình ảnh: Mỗi hình ảnh có kích thước nhỏ, cố định là 32x32 điểm ảnh với 3 kênh màu RGB.
- Số lớp (labels): CIFAR-10 chứa 10 lớp (categories), mỗi lớp bao gồm 6,000 hình ảnh. Các lớp bao gồm:
- Máy bay (airplane), ô tô (automobile), chim (bird), mèo (cat), hươu (deer), chó (dog), ếch (frog), ngựa (horse), tàu thủy (ship), và xe tải (truck).

**CIFAR-10 được chia thành hai phần chính:**

- Dữ liệu huấn luyện (Training Set): 50,000 hình ảnh, được phân phối đều giữa 10 lớp (mỗi lớp 5,000 hình ảnh).
- Dữ liệu kiểm tra (Test Set): 10,000 hình ảnh, cũng được phân phối đều giữa các lớp (mỗi lớp 1,000 hình ảnh).

Mỗi hình ảnh được gán nhãn (label) là một trong 10 lớp, đảm bảo tính cân bằng và không có sự chồng chéo giữa các lớp.

**2.5.2. Lý do lựa chọn tập dữ liệu cho nghiên cứu**

CIFAR-10 gồm 60,000 hình ảnh với kích thước 32x32 điểm ảnh, giúp giảm thiểu yêu cầu về tài nguyên tính toán. Điều này đặc biệt hữu ích khi triển khai mô hình trên các nền tảng hạn chế như Google Colab, nơi tài nguyên phần cứng có thể bị giới hạn.

Tập dữ liệu được cân bằng hoàn hảo với 10 lớp đối tượng, mỗi lớp chứa số lượng hình ảnh tương đương. Sự cân bằng này đảm bảo rằng mô hình không bị thiên vị (bias) đối với bất kỳ lớp nào trong quá trình huấn luyện và kiểm tra. Hơn nữa, sự đa dạng trong các lớp (từ động vật, phương tiện đến các đối tượng tự nhiên) giúp đánh giá khả năng tổng quát hóa của mô hình trên các loại dữ liệu khác nhau.

CIFAR-10 là một tập dữ liệu tiêu chuẩn, được sử dụng rộng rãi trong cộng đồng nghiên cứu thị giác máy tính. Điều này giúp dễ dàng so sánh hiệu suất của mô hình CNN với các nghiên cứu trước đó, tạo cơ sở để đánh giá và cải tiến mô hình.

Tập dữ liệu này được tích hợp sẵn trong nhiều thư viện học sâu phổ biến như TensorFlow, PyTorch, và Keras, giúp tiết kiệm thời gian trong việc tải, chuẩn bị và tiền xử lý dữ liệu. Đây là yếu tố thuận lợi cho việc tập trung vào phát triển và tối ưu hóa mô hình.



## 2.6. Công cụ và thư viện hỗ trợ

### 2.6.1. Python và các thư viện TensorFlow và Pytorch

Python là ngôn ngữ lập trình phổ biến và mạnh mẽ, đặc biệt trong lĩnh vực học sâu (Deep Learning). Với cú pháp đơn giản, dễ hiểu cùng hệ sinh thái phong phú, Python đã trở thành lựa chọn hàng đầu để triển khai các mô hình Mạng Nơ-ron Tích chập (CNN). Ngôn ngữ này cung cấp hàng loạt thư viện và công cụ hỗ trợ trong quá trình xử lý dữ liệu, xây dựng mô hình, huấn luyện, và đánh giá, từ đó giúp nhà nghiên cứu tiết kiệm thời gian và tập trung vào việc tối ưu hóa mô hình.

Hai thư viện học sâu phổ biến nhất tích hợp trên Python là TensorFlow và PyTorch. TensorFlow, được phát triển bởi Google, là một nền tảng mạnh mẽ với khả năng triển khai trên nhiều môi trường như CPU, GPU, hoặc TPU. Thư viện này cung cấp API thân thiện, đặc biệt là Keras, giúp việc thiết kế và huấn luyện các mô hình CNN trở nên dễ dàng hơn. Một tính năng nổi bật khác của TensorFlow là TensorBoard, công cụ trực quan hóa mạnh mẽ, cho phép theo dõi quá trình huấn luyện và tối ưu hóa mô hình. Điều này rất hữu ích trong việc phân tích hiệu suất và cải thiện kiến trúc CNN.

PyTorch, do Facebook phát triển, nổi bật với sự linh hoạt và khả năng tính toán động (dynamic computation graph). Điều này cho phép nhà nghiên cứu kiểm tra và thay đổi kiến trúc mô hình một cách dễ dàng trong quá trình triển khai. PyTorch còn tích hợp thư viện torchvision, cung cấp sẵn các tập dữ liệu phổ biến như CIFAR-10, cũng như các công cụ tiền xử lý dữ liệu mạnh mẽ. Tính năng hỗ trợ GPU hiệu quả của PyTorch giúp giảm thời gian huấn luyện, đặc biệt khi làm việc với các mô hình lớn hoặc tập dữ liệu phức tạp.

Trong bối cảnh của nghiên cứu này, cả TensorFlow và PyTorch đều là lựa chọn lý tưởng để triển khai mô hình CNN trên tập dữ liệu CIFAR-10. Với CIFAR-10, cả hai thư viện đều cung cấp các công cụ hỗ trợ tải dữ liệu, chuẩn hóa và tăng cường dữ liệu, cũng như xây dựng kiến trúc CNN từ cơ bản đến phức tạp. Khả năng tích hợp dễ dàng và tài liệu hướng dẫn phong phú giúp việc ứng dụng hai thư viện này trong nghiên cứu không chỉ thuận tiện mà còn đảm bảo kết quả chính xác và đáng tin cậy.

### 2.6.2. Nền tảng Google Colab và các tiện ích tích hợp

Google Colab (Colaboratory) là một nền tảng trực tuyến được phát triển bởi Google, cung cấp môi trường lập trình Python mạnh mẽ với khả năng sử dụng GPU/TPU miễn phí. Colab được thiết kế để hỗ trợ nghiên cứu và phát triển các mô hình học máy

(Machine Learning) và học sâu (Deep Learning), đặc biệt là trong các dự án sử dụng các thư viện như TensorFlow và PyTorch.

Google Colab hoạt động dựa trên giao diện Jupyter Notebook, cho phép người dùng thực hiện lập trình, ghi chú, và trực quan hóa trong cùng một tài liệu. Điều này giúp nhà nghiên cứu dễ dàng theo dõi, kiểm tra và trình bày kết quả trong quá trình triển khai mô hình CNN. Một ưu điểm nổi bật của Colab là khả năng chạy trực tiếp trên đám mây, loại bỏ nhu cầu cấu hình phần cứng phức tạp và tiết kiệm tài nguyên máy tính cá nhân.



Hình 2.6 Minh họa logo của Google Colab

## 2.7. Đánh giá hiệu suất mô hình

### 2.7.1. Các chỉ số đánh giá hiệu suất chính trong phân loại hình ảnh

Trong các bài toán phân loại hình ảnh, việc đánh giá hiệu suất mô hình không chỉ đơn thuần dựa trên số lượng dự đoán đúng, mà cần sử dụng các chỉ số toàn diện để hiểu rõ khả năng hoạt động của mô hình trên tập dữ liệu huấn luyện và kiểm tra. Các chỉ số này không chỉ giúp xác định mức độ chính xác của mô hình mà còn cung cấp cái nhìn sâu hơn về cách mô hình xử lý dữ liệu.

**Độ chính xác (Accuracy):** Đây là chỉ số cơ bản nhất, cho biết tỷ lệ các mẫu được dự đoán đúng so với tổng số mẫu trong tập dữ liệu. Độ chính xác thường được sử dụng khi dữ liệu cân bằng, tức là các lớp có số lượng mẫu tương đồng. Tuy nhiên, với dữ liệu không cân bằng, độ chính xác không phản ánh đúng hiệu suất của mô hình, vì mô hình có thể đạt độ chính xác cao mà vẫn bỏ sót các lớp nhỏ.

**Ma trận nhầm lẫn (Confusion Matrix):** Ma trận nhầm lẫn là công cụ quan trọng để đánh giá hiệu suất của mô hình trên từng lớp. Nó hiển thị số lượng dự đoán đúng và nhầm lẫn giữa các lớp trong bài toán phân loại. Từ ma trận này, ta có thể xác định mô hình dự đoán tốt ở những lớp nào và nhầm lẫn ở đâu. Điều này giúp phân tích sâu hơn về hiệu suất mô hình thay vì chỉ nhìn vào tổng thể.

**Precision và Recall:** Precision (độ chính xác của dự đoán) và Recall (độ nhạy) là hai chỉ số thường được sử dụng trong bài toán phân loại. Precision cho biết trong số

các mẫu mà mô hình dự đoán thuộc một lớp, bao nhiêu mẫu thực sự đúng. Trong khi đó, Recall đo lường khả năng phát hiện các mẫu thực sự thuộc về một lớp cụ thể. Hai chỉ số này rất quan trọng khi tập dữ liệu không cân bằng hoặc khi một số lớp có ý nghĩa quan trọng hơn các lớp khác.

**F1-Score:** F1-Score là chỉ số tổng hợp của Precision và Recall, được sử dụng khi cần đánh giá hiệu suất một cách cân bằng giữa độ chính xác và độ nhạy. Chỉ số này đặc biệt hữu ích khi hai chỉ số Precision và Recall có sự chênh lệch đáng kể, cung cấp một góc nhìn toàn diện hơn về hiệu suất của mô hình.

**Top-k Accuracy:** Đây là chỉ số đánh giá hiệu suất trong các bài toán phân loại có nhiều lớp. Thay vì chỉ xem xét nhãn dự đoán chính xác nhất, Top-k Accuracy kiểm tra xem nhãn đúng có nằm trong top k nhãn được mô hình dự đoán hay không. Chỉ số này thường được áp dụng trong các bài toán phức tạp, nơi độ chính xác tuyệt đối khó đạt được.

**Log-Loss (Hàm mất mát Log):** Log-Loss đo lường sự khác biệt giữa xác suất mà mô hình dự đoán và nhãn thực tế. Chỉ số này không chỉ phản ánh việc dự đoán đúng hay sai mà còn đánh giá mức độ tự tin trong dự đoán của mô hình. Giá trị Log-Loss thấp cho thấy mô hình không chỉ dự đoán đúng mà còn có mức độ chắc chắn cao.

### 2.7.2. Phương pháp phân tích kết quả huấn luyện

Phân tích kết quả huấn luyện là bước quan trọng để đánh giá hiệu suất mô hình CNN và đưa ra những điều chỉnh phù hợp nhằm cải thiện hiệu quả phân loại. Quá trình này tập trung vào việc theo dõi các chỉ số hiệu suất, phân tích chi tiết các lỗi dự đoán, và so sánh giữa dữ liệu huấn luyện, kiểm tra, và dữ liệu thực tế.

**Phân tích Loss và Accuracy qua từng epoch:** Loss thể hiện mức độ sai lệch giữa dự đoán của mô hình và nhãn thực tế. Dựa vào biểu đồ Loss trên tập huấn luyện và kiểm tra, có thể xác định liệu mô hình có đang bị quá khớp hay không. Accuracy đo lường tỷ lệ dự đoán đúng của mô hình và thường được theo dõi đồng thời với Loss. Sự hội tụ lý tưởng là khi Loss giảm và Accuracy tăng trên cả hai tập dữ liệu.

**Ma trận nhầm lẫn (Confusion Matrix):** Ma trận nhầm lẫn là công cụ giúp hiểu rõ cách mô hình phân loại các lớp trong tập dữ liệu. Nó cung cấp số liệu cụ thể về số lượng dự đoán đúng và sai giữa các lớp. Phân tích ma trận nhầm lẫn giúp nhận diện các lớp mà mô hình gặp khó khăn trong việc phân biệt. Từ đó có thể bổ sung dữ liệu hoặc tinh chỉnh mô hình để cải thiện hiệu suất.

**Phân tích Precision, Recall và F1-Score:** Precision đo lường tỷ lệ chính xác trong các dự đoán thuộc một lớp cụ thể. Recall đánh giá khả năng phát hiện đúng các mẫu thực sự thuộc về lớp đó. F1-Score kết hợp cả Precision và Recall để cung cấp cái nhìn cân bằng về hiệu suất của mô hình trên từng lớp. Các chỉ số này đặc biệt hữu ích trong các tập dữ liệu không cân bằng.

Việc phân tích không chỉ giúp đánh giá hiệu suất hiện tại của mô hình mà còn chỉ ra các điểm cần cải thiện. Từ các biểu đồ, ma trận nhầm lẫn và hình ảnh dự đoán sai, nhà nghiên cứu có thể xác định các lỗi hệ thống, cải thiện dữ liệu, hoặc tối ưu hóa cấu trúc mô hình. Phương pháp này đảm bảo mô hình đạt hiệu suất cao nhất trên cả tập dữ liệu huấn luyện và thực tế.

## 2.8. So sánh CNN với các mô hình học sâu khác

Convolutional Neural Network (CNN) là mô hình lý tưởng cho dữ liệu hình ảnh nhờ khả năng tự động trích xuất đặc trưng. Dưới đây là so sánh với các mô hình phổ biến khác:

- **DNN (Deep Neural Network):** Mạng truyền thống với các lớp kết nối đầy đủ, yêu cầu trích xuất đặc trưng thủ công, trong khi CNN tự động thực hiện.
- **RNN (Recurrent Neural Network):** Phù hợp cho dữ liệu chuỗi thời gian, trong khi CNN ưu tiên xử lý dữ liệu không tuần tự như hình ảnh.
- **ViT (Vision Transformer):** Dùng self-attention để học mối quan hệ xa trong ảnh, yêu cầu dữ liệu lớn hơn, trong khi CNN hiệu quả hơn trên tập dữ liệu nhỏ và trung bình.
- **ResNet (Residual Network):** Là biến thể cải tiến của CNN với kết nối tắt (skip connections) giúp học sâu hơn mà không làm mất thông tin.
- **EfficientNet:** Tối ưu hóa kích thước và hiệu suất tính toán, là bước tiến tiếp theo trong việc cải tiến CNN.

CNN là lựa chọn lý tưởng cho các bài toán nhận dạng đối tượng nhờ khả năng trích xuất đặc trưng tự động, hiệu quả và phù hợp với nhiều tập dữ liệu khác nhau.

## CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

### 3.1. Mô tả bài toán

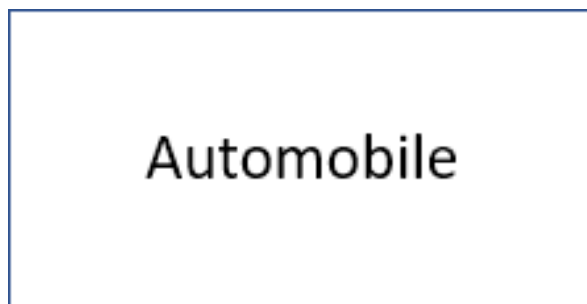
Bài toán được thực hiện nhằm giải quyết vấn đề nhận dạng đối tượng trong hình ảnh thông qua việc sử dụng mô hình Convolutional Neural Network (CNN), một phương pháp tiên tiến và phổ biến trong lĩnh vực thị giác máy tính. Mục tiêu chính là phân loại chính xác các đối tượng xuất hiện trong ảnh đầu vào, thuộc các danh mục khác nhau trong tập dữ liệu CIFAR-10. Tập dữ liệu này bao gồm 10 lớp đối tượng như máy bay, xe hơi, chim, mèo, hươu, chó, ếch, ngựa, tàu thủy và xe tải.

Hình ảnh đầu vào (Input) có kích thước 32x32 điểm ảnh với ba kênh màu RGB, thuộc tập dữ liệu CIFAR-10, được trình bày tương tự như Hình 3.1 dưới đây:



Hình 3.1 Minh họa ảnh đầu vào (Input)

Kết quả đầu ra (Output) của bài toán là một nhãn tương ứng với đối tượng trong hình ảnh đầu vào. Sau khi thực hiện các bước tiền xử lý và huấn luyện, mô hình sẽ phân loại và đưa ra kết quả như minh họa tương tự như Hình 3.2 dưới đây:



Hình 3.2. Minh họa kết quả đầu ra (Output) là một nhãn dự đoán

## 3.2. Tiền xử lý dữ liệu

### 3.2.1. Tải và phân tích dữ liệu CIFAR-10 với TensorFlow

Trong bước đầu tiên của quá trình xử lý dữ liệu, tập dữ liệu CIFAR-10 được tải và phân tích thông qua thư viện TensorFlow. Tập dữ liệu này bao gồm 60.000 hình ảnh với kích thước 32x32 điểm ảnh, được phân loại thành 10 danh mục, mỗi danh mục chứa 6.000 hình ảnh.

Để chuẩn bị dữ liệu CIFAR-10 cho quá trình huấn luyện mô hình, các bước xử lý dữ liệu được thực hiện như sau:

#### Tải dữ liệu:

- Sử dụng hàm `cifar10.load_data()` từ TensorFlow để tải trực tiếp và phân chia dữ liệu thành hai phần:
  - **Tập huấn luyện:** 50.000 hình ảnh và nhãn dùng để xây dựng mô hình.
  - **Tập kiểm tra:** 10.000 hình ảnh và nhãn dùng để đánh giá mô hình.

#### Kiểm tra kích thước dữ liệu:

- Sau khi tải thành công, kích thước của tập dữ liệu được kiểm tra để đảm bảo dữ liệu đã sẵn sàng:
  - **Tập huấn luyện:** (50000, 32, 32, 3) với nhãn (50000, 1).
  - **Tập kiểm tra:** (10000, 32, 32, 3) với nhãn (10000, 1).

#### Phân tích dữ liệu ban đầu:

- Kiểm tra số lượng hình ảnh giữa các lớp được phân bổ đều, đảm bảo sự cân bằng trong dữ liệu.

Sau khi hoàn tất bước tải và phân tích, dữ liệu CIFAR-10 đã được chuẩn bị đầy đủ, sẵn sàng cho các bước xử lý tiếp theo trong quy trình huấn luyện mô hình CNN.

### 3.2.2. Trực quan hóa dữ liệu bằng Seaborn

Trong quá trình xử lý dữ liệu ban đầu, việc trực quan hóa dữ liệu đóng vai trò quan trọng trong việc phân tích đặc điểm và phân phối của tập dữ liệu CIFAR-10. Ở bước này, thư viện Seaborn được sử dụng để tạo ra các biểu đồ trực quan, giúp nhận diện nhanh chóng sự phân bố của dữ liệu trong từng lớp.

Các bước xử lý được thực hiện như sau:

#### Tạo biểu đồ phân phối tập dữ liệu:

- Sử dụng hàm `barplot()` của Seaborn để hiển thị số lượng hình ảnh trong mỗi lớp của tập dữ liệu huấn luyện và kiểm tra.

- Số lượng hình ảnh của từng lớp được tính toán bằng cách đếm các nhãn tương ứng trong tập huấn luyện và tập kiểm tra.

- **Hiện thị thông tin qua biểu đồ:**

- Biểu đồ phân phối được tạo cho cả tập huấn luyện và tập kiểm tra.
- Hai biểu đồ được đặt cạnh nhau với kích thước `figsize=(12, 4)` để so sánh trực quan và đánh giá sự cân bằng dữ liệu giữa các lớp.

Việc trực quan hóa này cung cấp cái nhìn tổng quan rõ ràng về sự phân bố dữ liệu giữa các lớp, từ đó hỗ trợ kiểm tra tính cân bằng và chuẩn bị dữ liệu hiệu quả cho các bước xử lý tiếp theo trong quy trình huấn luyện mô hình CNN.

### 3.2.3. Chuẩn hóa dữ liệu và mã hóa nhãn bằng One-Hot Encoding

Trong quá trình xử lý dữ liệu, việc chuẩn hóa và mã hóa nhãn là bước quan trọng để đảm bảo dữ liệu đồng nhất và phù hợp với đầu vào của mô hình CNN, giúp chuyển đổi nhãn sang định dạng tương thích với các thuật toán học sâu:

- **Chuẩn hóa dữ liệu:**

- Ảnh trong tập dữ liệu được chuẩn hóa về khoảng giá trị  $[0, 1]$  bằng cách chia toàn bộ giá trị điểm ảnh cho 255.
- Điều này giúp giảm độ lớn của dữ liệu, tối ưu hóa hiệu suất tính toán và tăng khả năng hội tụ trong quá trình huấn luyện.

- **Mã hóa nhãn bằng One-Hot Encoding:**

- Nhãn của mỗi hình ảnh trong tập dữ liệu ban đầu được biểu diễn dưới dạng số nguyên từ 0 đến 9, tương ứng với các danh mục.
- Nhãn này được chuyển đổi sang dạng ma trận nhị phân bằng cách sử dụng hàm `tf.keras.utils.to_categorical()`.
- Quá trình mã hóa này đảm bảo rằng mỗi nhãn được biểu diễn trong không gian 10 chiều, tương ứng với 10 danh mục.

Việc chuẩn hóa dữ liệu và mã hóa nhãn đảm bảo đầu vào mô hình CNN tuân thủ các yêu cầu chuẩn hóa của thuật toán.

### 3.2.4. Tăng cường dữ liệu bằng ImageDataGenerator

Trong quá trình xử lý dữ liệu cuối cùng này, việc tăng cường dữ liệu nhằm làm phong phú tập dữ liệu huấn luyện và cải thiện khả năng tổng quát hóa của mô hình CNN. Kỹ thuật này giúp giảm thiểu hiện tượng quá khớp (overfitting) bằng cách tạo ra các biến thể mới từ dữ liệu, sử dụng thư viện ImageDataGenerator của TensorFlow và Keras.

Các bước xử lý được thực hiện như sau:

- **Cấu hình lớp ImageDataGenerator:**

- **Xoay ảnh:** Tạo ra ảnh xoay trong khoảng  $\pm 5$  độ.
- **Dịch chuyển ngang:** Dịch chuyển ảnh trong phạm vi  $\pm 5\%$  chiều rộng.
- **Dịch chuyển dọc:** Dịch chuyển ảnh trong phạm vi  $\pm 5\%$  chiều cao.
- **Biến dạng hình học:** Áp dụng biến dạng với độ lệch tối đa 0.05 (5%).
- **Phóng to/thu nhỏ:** Thay đổi kích thước ảnh ngẫu nhiên khoảng  $\pm 5\%$ .
- **Lật ảnh ngang:** Thực hiện lật ảnh ngẫu nhiên theo chiều ngang.
- **Lấp đầy vùng trống:** Sử dụng phương pháp "nearest" (gần nhất) để lấp đầy các vùng trống phát sinh sau các phép biến đổi.

- **Thực hiện tăng cường dữ liệu:**

- Phương thức `fit()` được sử dụng trên các tập dữ liệu huấn luyện (`x_train`) để tính toán các thông số chuẩn hóa cần thiết.
- Trong quá trình huấn luyện, dữ liệu tăng cường được tạo ra tự động thông qua generator (`datagen.flow()`), đảm bảo việc cung cấp dữ liệu mới một cách liên tục.

- **Hiển thị mẫu dữ liệu tăng cường:**

- Chọn ngẫu nhiên một số hình ảnh gốc từ tập dữ liệu để so sánh với ảnh sau khi thực hiện tăng cường dữ liệu và hiển thị thông qua phương thức `random_transform()`.

### 3.3. Xây dựng kiến trúc mô hình CNN

#### 3.3.1. Thiết kế kiến trúc CNN

Trong quá trình xây dựng mô hình Convolutional Neural Network (CNN), việc thiết kế kiến trúc đóng vai trò quan trọng nhằm đảm bảo khả năng học tập và phân loại chính xác cho dữ liệu hình ảnh CIFAR-10. Các lớp trong mô hình được sắp xếp theo thứ tự cụ thể và lựa chọn cẩn thận các thông số nhằm tối ưu hóa quá trình học.

Dưới đây là các thông số chi tiết của từng lớp trong kiến trúc CNN được đánh số thứ tự theo quy trình:

**1. Ảnh đầu vào:** Kích thước: 32x32x3 (RGB với 3 kênh màu).

**2. Khối Block 1:**

- **Lớp tích chập 1:**

- **Số bộ lọc:** 32.



- **Kích thước bộ lọc:** 3x3.
- **Padding:** same (giữ nguyên kích thước ảnh sau khi chập).
- **Hàm kích hoạt:** ReLU.
- **Regularization:** L2 với hệ số 0.001.
- **Lớp tích chập 1.2:**
  - **Số bộ lọc:** 64.
  - **Kích thước bộ lọc:** 3x3.
  - **Hàm kích hoạt:** ReLU.
  - **Regularization:** L2 với hệ số 0.001.
- **Lớp gộp 1:**
  - **Kích thước kernel:** 2x2.
  - **Phương pháp:** Max Pooling.
- **Dropout:**
  - **Tỷ lệ dropout:** 0.25 (giảm nguy cơ quá khớp).

### 3. Khối Block 2:

- **Lớp tích chập 2:**
  - **Số bộ lọc:** 64.
  - **Kích thước bộ lọc:** 3x3.
  - **Padding:** same.
  - **Hàm kích hoạt:** ReLU.
  - **Regularization:** L2 với hệ số 0.001.
- **Lớp tích chập 2.2:**
  - **Số bộ lọc:** 128.
  - **Kích thước bộ lọc:** 3x3.
  - **Hàm kích hoạt:** ReLU.
  - **Regularization:** L2 với hệ số 0.001.
- **Lớp gộp 2:**
  - **Kích thước kernel:** 2x2.
  - **Phương pháp:** Max Pooling.
- **Dropout:**
  - **Tỷ lệ dropout:** 0.25.

### 4. Khối Block 3:

- **Lớp tích chập 3:**
  - **Số bộ lọc:** 128.
  - **Kích thước bộ lọc:** 3x3.
  - **Padding:** same.
  - **Hàm kích hoạt:** ReLU.
  - **Regularization:** L2 với hệ số 0.001.
- **Lớp tích chập 3.2:**
  - **Số bộ lọc:** 256.
  - **Kích thước bộ lọc:** 3x3.
  - **Hàm kích hoạt:** ReLU.
  - **Regularization:** L2 với hệ số 0.001.
- **Lớp gộp 3:**
  - **Kích thước kernel:** 2x2.
  - **Phương pháp:** Max Pooling.
- **Dropout:**
  - **Tỷ lệ dropout:** 0.25.

#### 5. Lớp kết nối đầy đủ (Fully Connected):

- **Lớp Flatten:** Chuyển tensor đầu ra từ lớp tích chập thành dạng vector.
- **Lớp Dense:**
  - **Số đơn vị:** 512.
  - **Hàm kích hoạt:** ReLU.
  - **Regularization:** L2 với hệ số 0.001.
- **Dropout:**
  - **Tỷ lệ dropout:** 0.4.

#### 6. Lớp đầu ra:

- **Số đơn vị:** 10 (tương ứng với 10 lớp của CIFAR-10).
- **Hàm kích hoạt:** Softmax.

#### Vai trò của từng khối trong kiến trúc:

**Block 1 - Block 3:** Các khối này là nền tảng của mô hình CNN, thực hiện việc xử lý và trích xuất các đặc điểm từ hình ảnh đầu vào. Mỗi khối tập trung vào việc phát hiện và biểu diễn các mức độ đặc điểm khác nhau:

- **Block 1:**

- Đây là bước đầu tiên trong quá trình trích xuất đặc điểm từ tập dữ liệu hình ảnh.
- Các lớp tích chập trong Block 1 tập trung vào việc nhận diện các đặc điểm cơ bản, như cạnh, góc và các chi tiết đơn giản.
- Việc sử dụng bộ lọc kích thước 3x3 với padding giúp giữ nguyên kích thước hình ảnh, đảm bảo không làm mất dữ liệu quan trọng.
- Max Pooling 2x2 giảm kích thước đặc trưng, đồng thời giữ lại các đặc điểm quan trọng, trong khi Dropout với tỷ lệ 0.25 giúp tránh hiện tượng quá khớp.

- **Block 2:**

- Các lớp tích chập trong Block 2 tiếp tục xử lý và phân tích các đặc điểm phức tạp hơn từ kết quả của Block 1.
- Block 2 sử dụng số lượng bộ lọc lớn hơn (64 và 128), giúp phát hiện các mẫu trung cấp, như hình dạng hoặc cấu trúc trong hình ảnh.
- Việc sử dụng Max Pooling và Dropout tiếp tục giảm thiểu kích thước và ngăn quá khớp, đảm bảo mô hình học hiệu quả mà không bị nhiễu bởi các đặc điểm không quan trọng.

- **Block 3:**

- Đây là bước cuối cùng trong quá trình trích xuất đặc điểm, các lớp trong Block 3 tập trung vào việc học các đặc điểm sâu hơn và cụ thể hơn, chẳng hạn như các mẫu phức tạp đại diện cho từng lớp trong tập dữ liệu.
- Với số lượng bộ lọc lớn hơn (128 và 256), các lớp trong Block 3 có khả năng phân tích sâu và học các đặc điểm tinh vi của dữ liệu.
- Tương tự như các block trước, Max Pooling và Dropout tiếp tục được sử dụng để giảm thiểu kích thước và tránh hiện tượng quá khớp.

**Lớp Fully Connected (Kết nối đầy đủ):** Đảm nhiệm vai trò tổng hợp các đặc điểm đã trích xuất từ các block trước và thực hiện phân loại.

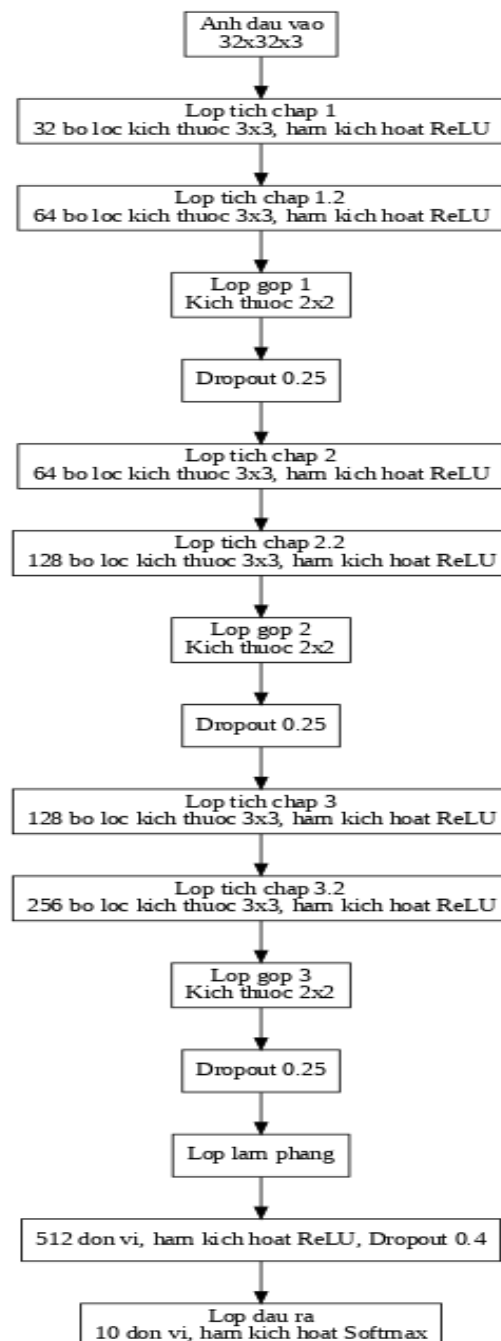
- Lớp Flatten chuyển đổi dữ liệu từ dạng tensor 3 chiều thành vector 1 chiều để làm đầu vào cho các lớp Dense.

- Lớp Dense với 512 đơn vị học đặc điểm tổng quát và lớp cuối cùng (Softmax) dự đoán xác suất thuộc về mỗi lớp trong 10 lớp của tập dữ liệu CIFAR-10.

**Dropout:** Được chèn sau mỗi khối tích chập và lớp kết nối đầy đủ để giảm thiểu hiện tượng quá khớp (overfitting).

**Regularization:** L2 được áp dụng để thêm các ràng buộc, giảm thiểu việc mô hình trở nên quá phức tạp, đảm bảo khả năng tổng quát hóa trên dữ liệu chưa thấy trước.

Để hiểu rõ hơn về cách các lớp trong mô hình CNN được tổ chức và liên kết với nhau, hình 3.3. dưới đây trình bày lưu đồ chi tiết về kiến trúc của mô hình, được minh họa tương ứng:



Hình 3.2 Minh họa lưu đồ kiến trúc CNN

### 3.3.2. Triển khai mô hình CNN

Sau khi thiết kế kiến trúc, bước tiếp theo là triển khai mô hình CNN trên dữ liệu CIFAR-10. Quá trình triển khai bao gồm việc cấu hình các thông số quan trọng như hàm mất mát, bộ tối ưu hóa, và các chỉ số đánh giá, đồng thời thực hiện việc huấn luyện trên tập dữ liệu huấn luyện.

Dưới đây là các bước chi tiết trong quá trình triển khai:

#### 1. Cấu hình mô hình

Trước khi bắt đầu huấn luyện, mô hình được cấu hình với các tham số sau:

- **Hàm mất mát (Loss Function):** Sử dụng `categorical_crossentropy` phù hợp với bài toán phân loại đa lớp, giúp mô hình tối ưu hóa việc dự đoán xác suất chính xác.
- **Bộ tối ưu hóa (Optimizer):** Sử dụng RMSprop với `learning rate = 0.0001`. Bộ tối ưu hóa này giúp điều chỉnh trọng số của mô hình một cách hiệu quả trong không gian tham số.
- **Các chỉ số đánh giá (Metrics):** Độ chính xác (accuracy) được sử dụng để đánh giá mức độ mô hình dự đoán chính xác các lớp.

#### 2. Huấn luyện mô hình

Mô hình CNN được huấn luyện trên tập dữ liệu CIFAR-10 thông qua hai phương pháp chính:

- **Không sử dụng tăng cường dữ liệu:**
  - Dữ liệu huấn luyện trực tiếp mà không áp dụng các biến đổi.
  - Sử dụng `batch_size = 32` và `epochs = 100` để kiểm soát quá trình huấn luyện.
- **Sử dụng tăng cường dữ liệu (Data Augmentation):**
  - Tăng cường dữ liệu bằng ImageDataGenerator, tạo các biến thể mới từ dữ liệu ban đầu bằng cách áp dụng các biến đổi như xoay, dịch chuyển, phóng to, thu nhỏ, và lật ảnh.
  - Điều này giúp mô hình học được các đặc điểm đa dạng, từ đó tăng khả năng tổng quát hóa.
  - Sử dụng EarlyStopping và ReduceLROnPlateau để ngăn ngừa hiện tượng quá khớp và tối ưu hóa tốc độ học.

Việc triển khai mô hình CNN bao gồm các bước từ cấu hình, huấn luyện, theo dõi hiệu suất, đến lưu trữ mô hình. Bằng cách sử dụng tăng cường dữ liệu và các kỹ thuật tối ưu hóa như EarlyStopping, mô hình có khả năng học hiệu quả và giảm thiểu nguy cơ quá khớp, từ đó đạt được độ chính xác cao trên tập dữ liệu kiểm tra.

### 3.4. Đánh giá hiệu suất mô hình

Sau khi mô hình Convolutional Neural Network (CNN) đã được huấn luyện trên tập dữ liệu CIFAR-10, bước tiếp theo là đánh giá hiệu suất để kiểm tra khả năng tổng quát hóa của mô hình trên tập dữ liệu kiểm tra. Việc đánh giá này bao gồm nhiều khía cạnh, từ các chỉ số cơ bản như độ chính xác và hàm mất mát đến các phân tích chi tiết hơn như ma trận nhầm lẫn và các dự đoán sai.

#### 1. Đánh giá bằng các chỉ số.

- Sử dụng các hàm đánh giá hiệu suất do thư viện TensorFlow và Keras cung cấp, chúng ta có thể thu thập được các chỉ số quan trọng sau:
  - **Loss:** Được tính toán bằng cách sử dụng hàm `model.evaluate()` của Keras, đây là thước đo sự sai lệch giữa dự đoán của mô hình và nhãn thực tế của tập dữ liệu.
  - **Accuracy:** Giá trị này cũng được trả về từ hàm `model.evaluate()`, đồng thời, còn thể hiện tỷ lệ phần trăm số mẫu dự đoán đúng trong tổng số mẫu kiểm tra.

#### 2. Ma trận nhầm lẫn (Confusion Matrix)

Ma trận nhầm lẫn được sử dụng để trình bày chi tiết hiệu suất của mô hình trên từng lớp cụ thể. Đây là một bảng 2 chiều, trong đó:

- Hàng dọc đại diện cho các nhãn thực tế (True Labels).
- Hàng ngang đại diện cho các nhãn dự đoán (Predicted Labels).

Mỗi ô trong ma trận nhầm lẫn cho biết số lượng mẫu thuộc một nhãn thực tế được mô hình dự đoán vào một nhãn cụ thể.

#### 3. Báo cáo phân loại (Classification Report)

Báo cáo phân loại cung cấp cái nhìn chi tiết hơn về hiệu suất của mô hình trên từng lớp bằng các chỉ số:

- **Precision (Độ chính xác):** Tỷ lệ dự đoán đúng trong số các mẫu được mô hình dự đoán thuộc lớp đó. Precision cao cho thấy mô hình ít nhầm lẫn giữa lớp này với các lớp khác.

- **Recall (Độ nhạy):** Tỷ lệ dự đoán đúng trong số các mẫu thực sự thuộc lớp đó. Recall cao cho thấy mô hình có khả năng phát hiện các mẫu của lớp đó tốt hơn.
- **F1-Score:** Là trung bình điều hòa của Precision và Recall, cung cấp một chỉ số cân bằng giữa hai giá trị này.

#### 4. Phân tích các lỗi

Việc phân tích các mẫu dự đoán sai đóng vai trò quan trọng trong việc cải thiện mô hình. Bằng cách xác định các mẫu mà mô hình dự đoán sai, chúng ta có thể tìm hiểu lý do:

- **Dữ liệu không rõ ràng:** hình ảnh của "Bird" có thể bị nhầm lẫn với "Airplane" nếu chúng có bối cảnh giống nhau (như bầu trời).
- **Lớp có đặc điểm tương tự:** Một số lớp như "Dog" và "Cat" có các đặc điểm hình ảnh tương tự, dẫn đến sự nhầm lẫn.
- **Dữ liệu không cân bằng:** Một số lớp có ít mẫu hơn, khiến mô hình học không đủ thông tin để phân biệt chính xác.

#### 5. Trục quan hóa hiệu suất huấn luyện

Hiệu suất của mô hình qua từng epoch được thể hiện qua hai biểu đồ:

- **Accuracy (Độ chính xác):** Hiển thị sự cải thiện của độ chính xác trên tập huấn luyện và kiểm tra qua từng epoch.
- **Loss (Hàm mất mát):** Hiển thị sự giảm dần của hàm mất mát qua từng epoch.

## CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

### 4.1. Kết quả huấn luyện

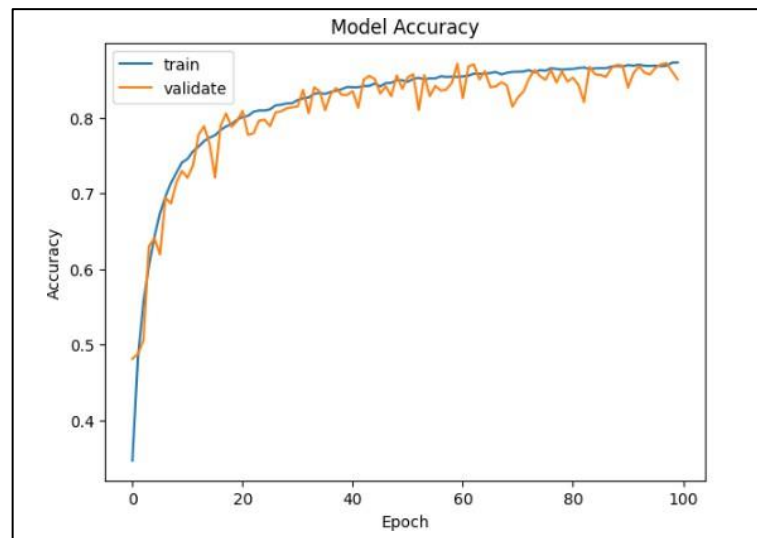
Sau quá trình huấn luyện mô hình Convolutional Neural Network (CNN) với tập dữ liệu CIFAR-10, hai chỉ số quan trọng là Loss (hàm mất mát) và Accuracy (độ chính xác) được theo dõi và ghi nhận qua từng epoch. Việc theo dõi này giúp đánh giá khả năng học tập và tổng quát hóa của mô hình trong từng giai đoạn huấn luyện.

#### Biểu đồ Accuracy qua từng epoch:

- Đường màu xanh thể hiện Accuracy trên tập dữ liệu huấn luyện.
- Đường màu đỏ thể hiện Accuracy trên tập dữ liệu kiểm tra.

#### Phân tích xu hướng trong biểu đồ:

- Độ chính xác trên tập huấn luyện tăng lên qua từng epoch, thể hiện rằng mô hình ngày càng nhận diện tốt hơn các mẫu trong tập dữ liệu này.
- Độ chính xác trên tập kiểm tra cũng tăng dần, đạt được mức độ cao nhất khi mô hình tiến tới hội tụ. Điều này chứng tỏ khả năng tổng quát hóa tốt của mô hình.



Hình 4.1 Minh họa biểu đồ độ chính xác (Accuracy)

Mô hình đạt độ chính xác 85% trên tập kiểm tra: Đây là tỷ lệ phần trăm mẫu trong tập kiểm tra được mô hình dự đoán chính xác. Với mức độ chính xác này, mô hình đã thể hiện khả năng học hiệu quả và có thể phân loại chính xác hầu hết các mẫu. Độ chính xác cao cũng phản ánh rằng các bước như tăng cường dữ liệu và điều chỉnh tham số đã phát huy tác dụng.



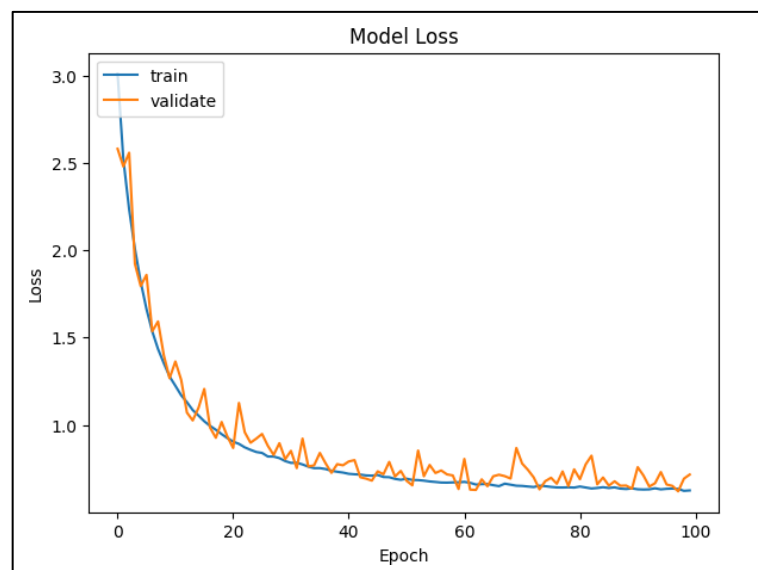
### Biểu đồ Loss qua từng epoch:

- Đường màu xanh thể hiện Loss trên tập dữ liệu huấn luyện.
- Đường màu đỏ thể hiện Loss trên tập dữ liệu kiểm tra.

### Phân tích xu hướng trong biểu đồ:

- Giá trị Loss trên tập huấn luyện giảm dần qua từng epoch, điều này cho thấy mô hình đang dần học được cách tối ưu hóa tham số để giảm sai số trong quá trình dự đoán.
- Giá trị Loss trên tập kiểm tra cũng giảm dần và tiến tới hội tụ, đây là một dấu hiệu tích cực, chứng minh rằng mô hình không chỉ học tốt trên dữ liệu huấn luyện mà còn tổng quát hóa được trên dữ liệu chưa từng thấy.

Mô hình đạt giá trị hàm mất mát là 0.71 trên tập kiểm tra: Hàm mất mát phản ánh mức độ sai lệch trung bình giữa dự đoán của mô hình và nhãn thực tế. Với giá trị 0.71, mô hình đã học được các đặc điểm quan trọng từ dữ liệu, cho thấy quá trình tối ưu hóa diễn ra hiệu quả



Hình 4.2 Minh họa biểu đồ hàm mất mát (Loss)

Sau khi hoàn thành quá trình huấn luyện với tập dữ liệu CIFAR-10, mô hình Convolutional Neural Network (CNN) đã được đánh giá trên tập kiểm tra để xác định khả năng dự đoán và phân loại. Các chỉ số quan trọng ghi nhận được như sau:

```
Test loss: 0.7166112065315247
Test accuracy: 0.8507000207901001
```

Hình 4.3 Minh họa hình ảnh kết quả sau khi huấn luyện

## 4.2 Dự đoán trên tập kiểm tra

### 4.2.1 Kết quả dự đoán chính xác

Để kiểm tra hiệu suất của mô hình CNN trên các hình ảnh chưa từng thấy trước đó, các mẫu từ tập kiểm tra đã được dự đoán và đối chiếu với nhãn thực tế. Dưới đây là hình ảnh minh họa cho những dự đoán chính xác:



Hình 4.4 Minh họa hình ảnh những đoán chính xác

Các trường hợp dự đoán chính xác cho thấy hiệu quả của kiến trúc mô hình trong việc xử lý và nhận diện đặc điểm dữ liệu từ các hình ảnh phức tạp. Điều này khẳng định rằng CNN đã học được các đặc điểm nổi bật từ quá trình huấn luyện và có khả năng tổng quát hóa tốt.

### 4.2.2 Kết quả dự đoán sai

Những trường hợp dự đoán sai minh họa khả năng của mô hình CNN trong việc nhận diện ảnh từ tập dữ liệu kiểm tra. Một số dự đoán sai có thể giải thích được bởi sự tương đồng đặc điểm giữa các lớp, trong khi một số khác đến từ nền nhiễu hoặc chất lượng hình ảnh thấp. Hình minh họa dưới đây thể hiện một số trường hợp dự đoán sai. Các nhãn thực tế và nhãn dự đoán được hiển thị kèm theo mỗi hình ảnh để trực quan hóa rõ ràng từng trường hợp:



Hình 4.5 Minh họa hình ảnh những dự đoán sai

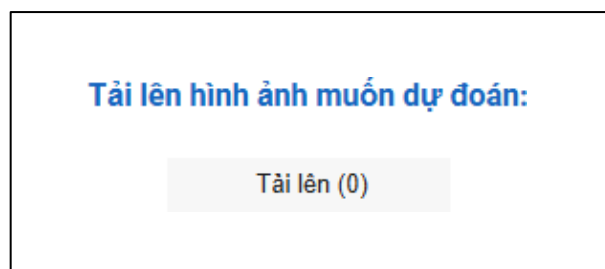
Những nhầm lẫn trên cho thấy cần cải thiện hơn nữa khả năng nhận diện đặc điểm của mô hình thông qua tăng cường dữ liệu hoặc điều chỉnh kiến trúc. Dữ liệu minh họa cung cấp góc nhìn cụ thể để tối ưu hóa hệ thống.

#### 4.2.3 Dự đoán hình ảnh thực tế

Mô hình CNN không chỉ giới hạn trong việc nhận dạng hình ảnh từ tập dữ liệu kiểm tra mà còn mở rộng khả năng nhận dạng hình ảnh thực tế do người dùng tải lên. Việc dự đoán những hình ảnh không có trong tập dữ liệu giúp kiểm tra tính hiệu quả của mô hình trong các tình huống thực tiễn, mang lại sự trực quan và tiện lợi thông qua một quy trình đơn giản.

##### Bước 1: Ấn vào nút "Tải lên"

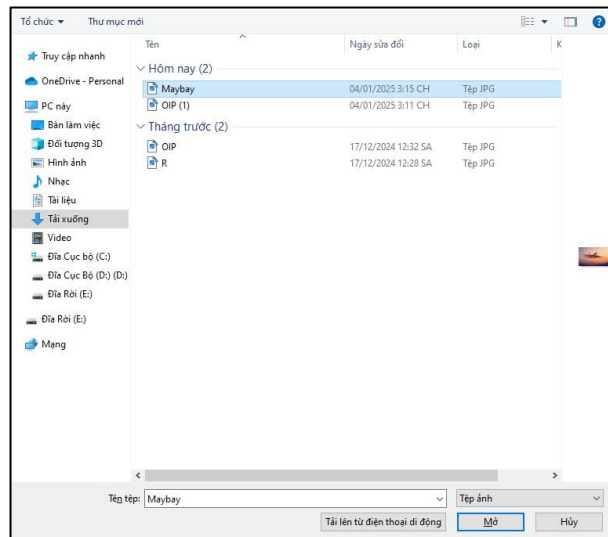
- Nhấp vào nút "Tải lên" xuất hiện trên giao diện.
- Đây là bước đầu tiên để tải lên hình ảnh muốn dự đoán



Hình 4.6 Minh họa hình ảnh chức năng tải ảnh lên

##### Bước 2: Chọn ảnh muốn dự đoán

- Từ cửa sổ hiện ra, điều hướng đến thư mục chứa hình ảnh muốn dự đoán.
- Chọn tệp ảnh và nhấn "Open" (Mở).



Hình 4.7 Minh họa hình ảnh chọn ảnh từ thư mục

### Bước 3: Tiến hành dự đoán

- Sau khi chọn ảnh, hệ thống tự động tiến hành xử lý và dự đoán kết quả.
- Trong thời gian này, sẽ thấy trạng thái "**Đang tiến hành dự đoán...**" hiển thị trên màn hình.



Hình 4.8 Minh họa hình ảnh đang tiến hành dự đoán

### Bước 4: Hiển thị kết quả dự đoán

- Kết quả dự đoán bao gồm:
  - Tên file ảnh:** Tên tệp hình ảnh bạn đã tải lên.
  - Kích thước gốc:** Kích thước ban đầu của hình ảnh.
  - Kích thước sau chuẩn hóa:** Kích thước ảnh sau khi được chuyển đổi để phù hợp với mô hình.
  - Lớp dự đoán:** Lớp mà mô hình đã dự đoán.
  - Ảnh đầu vào:** Hiển thị hình ảnh gốc đã được tải lên.



Hình 4.9 Minh họa hình ảnh kết quả dự đoán

Khả năng dự đoán chính xác các hình ảnh thực tế không thuộc tập dữ liệu đã huấn luyện thể hiện rõ năng lực tổng quát hóa vượt trội của mô hình CNN. Điều này không chỉ minh chứng cho hiệu quả của quá trình huấn luyện mà còn mở ra tiềm năng áp dụng mô hình vào các bài toán thực tiễn đa dạng, nơi mà dữ liệu đầu vào có thể thay đổi linh hoạt và không bị giới hạn bởi tập dữ liệu ban đầu.

### 4.3 Hiệu suất tổng thể

**Hiệu suất tốt:** Các lớp như Automobile (1), Horse (7), Ship (8), và Truck (9) đạt chỉ số F1-Score cao (từ 0.90 trở lên), phản ánh khả năng phân loại mạnh mẽ và ít nhầm lẫn của mô hình trên các lớp này.

**Hiệu suất trung bình:** Lớp Deer (4) và Frog (6) có Recall cao (tương ứng 0.90 và 0.97) nhưng Precision thấp hơn, cho thấy mô hình nhận diện tốt các mẫu của hai lớp này nhưng đôi khi nhầm lẫn chúng với các lớp khác.

**Hiệu suất kém:** Lớp Cat (3) có F1-Score thấp nhất (0.73), chủ yếu do Recall thấp (0.68). Điều này phản ánh mô hình gặp khó khăn khi phân loại các hình ảnh thuộc lớp này, có thể do sự tương đồng đặc điểm với các lớp như Dog (5).

**Cân bằng Precision và Recall:** Các lớp như Bird (2) và Ship (8) đạt cân bằng tốt giữa Precision và Recall, phản ánh hiệu suất ổn định.

	precision	recall	f1-score	support
0	0.91	0.81	0.86	1000
1	0.95	0.93	0.94	1000
2	0.88	0.78	0.83	1000
3	0.79	0.68	0.73	1000
4	0.78	0.90	0.84	1000
5	0.91	0.69	0.78	1000
6	0.71	0.97	0.82	1000
7	0.95	0.84	0.90	1000
8	0.86	0.95	0.90	1000
9	0.85	0.95	0.90	1000
accuracy			0.85	10000
macro avg	0.86	0.85	0.85	10000
weighted avg	0.86	0.85	0.85	10000

Tổng thể, các kết quả phân tích chỉ số Precision, Recall, và F1-Score cho thấy mô hình CNN hoạt động hiệu quả trên hầu hết các lớp, tuy nhiên vẫn còn một số hạn chế ở các lớp có đặc điểm tương đồng, đặc biệt là lớp Cat (3). Điều này gợi mở hướng cải tiến trong việc tăng cường dữ liệu hoặc tinh chỉnh kiến trúc mô hình để cải thiện hiệu suất trên các lớp khó phân loại.

## CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1. Kết luận

Đề tài đã hoàn thành việc xây dựng và đánh giá mô hình CNN trên tập dữ liệu CIFAR-10, đáp ứng các mục tiêu đề ra. Với độ chính xác 85% trên tập kiểm tra, mô hình thể hiện năng lực vượt trội trong việc học và phân loại hình ảnh. Đây là thành tựu đáng khích lệ, đặc biệt khi đối mặt với tập dữ liệu CIFAR-10 có tính đa dạng và độ phức tạp cao, mở ra triển vọng áp dụng trong các bài toán thị giác máy tính sau này.

### 5.2. Hạn chế

Mặc dù đạt được các kết quả khả quan, nghiên cứu này vẫn tồn tại một số hạn chế đáng chú ý:

- **Hiệu suất trên các lớp phức tạp:** Các lớp như Cat và Dog có chỉ số F1-Score thấp hơn so với các lớp khác, phản ánh sự nhầm lẫn của mô hình khi xử lý các đặc điểm tương đồng giữa các đối tượng.
- **Dữ liệu không đồng đều:** Mặc dù tập dữ liệu CIFAR-10 được phân bố đều, đặc điểm giữa các lớp vẫn có sự khác biệt đáng kể, gây khó khăn cho việc học các đặc điểm trừu tượng của mô hình.
- **Tính tổng quát hóa:** Khi áp dụng dự đoán trên hình ảnh thực tế ngoài tập dữ liệu huấn luyện, mô hình đôi khi đưa ra kết quả không chính xác, do chưa được tối ưu hóa để nhận dạng các đặc điểm phức tạp hơn.
- **Thời gian huấn luyện:** Với cấu hình phần cứng thông thường, việc huấn luyện mô hình với các thông số hiện tại tiêu tốn nhiều thời gian, đặc biệt khi số epoch cao hoặc sử dụng các phương pháp tăng cường dữ liệu.

### 5.3. Phương hướng phát triển

Nghiên cứu trong tương lai có thể tập trung vào các phương hướng dưới đây nhằm cải thiện mô hình và mở rộng tiềm năng ứng dụng:

- **Mở rộng và đa dạng hóa tập dữ liệu:** Áp dụng mô hình trên các tập dữ liệu lớn hơn hoặc các bài toán có độ phức tạp cao hơn, như nhận diện hình ảnh trong điều kiện ánh sáng kém, dữ liệu từ ảnh vệ tinh hoặc ảnh y tế.
- **Tối ưu thời gian huấn luyện:** Sử dụng các thuật toán tối ưu hóa tiên tiến như AdamW, Ranger, hoặc các chiến lược huấn luyện phân tán (Distributed Training) để giảm thiểu thời gian mà vẫn đảm bảo hiệu suất cao.
- **Ứng dụng thực tế và mở rộng phạm vi ứng dụng:** Ứng dụng mô hình CNN vào các hệ thống như nhận diện khuôn mặt, phân loại sản phẩm,...

## DANH MỤC TÀI LIỆU THAM KHẢO

- [1] T. Nguyễn, “Bizfly,” VCCorp, 26 10 2024. [Trực tuyến]. Available: bizfly.vn. [Đã truy cập 15 6 2006].
- [2] S. ngqolosi, “Medium,” ., 14 7 2023. [Trực tuyến]. Available: medium.com.
- [3] C. P. Van, “Viblo,” Azko, 10 12 2020. [Trực tuyến]. Available: viblo.asia.
- [4] phamdinhhkhanh, “phamdinhhkhanh,” ABC, 22 3 2019. [Trực tuyến]. Available: phamdinhhkhanh.github.io.
- [5] Pytorch, “Pytorch,” Pytorch, 12 6 2024. [Trực tuyến]. Available: pytorch.org.



## PHỤ LỤC