

TRƯỜNG ĐẠI HỌC TRÀ VINH  
TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CƠ SỞ NGÀNH**  
**HỌC KỲ I, NĂM HỌC 2025-2026**

**MÔ HÌNH CNN CHO PHÂN LOẠI ĐỘNG VẬT**

*Giảng viên hướng dẫn:*

Ths. Ngô Thanh Huy

*Sinh viên thực hiện:*

Họ tên: Nguyễn Hữu Tiến

MSSV: 110123185

Lớp: DA23TTD

Vĩnh Long, tháng 01 năm 2026

TRƯỜNG ĐẠI HỌC TRÀ VINH  
TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CƠ SỞ NGÀNH**  
**HỌC KỲ I, NĂM HỌC 2025-2026**

**MÔ HÌNH CNN CHO PHÂN LOẠI ĐỘNG VẬT**

*Giảng viên hướng dẫn:*

Ths. Ngô Thanh Huy

*Sinh viên thực hiện:*

Họ tên: Nguyễn Hữu Tiến

MSSV: 110123185

Lớp: DA23TTD

Vĩnh Long, tháng 01 năm 2026

**TRANG NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins or other markings on the paper.

Vĩnh Long, ngày ... tháng ... năm 2026

**Giảng viên hướng dẫn**

(Ký tên và ghi rõ họ tên)

**TRANG NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

*Vĩnh Long, ngày ... tháng ... năm 2026*

**Thành viên hội đồng**

*(Ký tên và ghi rõ họ tên)*

## LỜI CẢM ƠN

Em xin chân thành gửi lời cảm ơn sâu sắc đến quý thầy cô khoa Công nghệ Thông tin, đặc biệt là giảng viên hướng dẫn em – thầy Ngô Thanh Huy, người đã tận tình hướng dẫn, hỗ trợ và khơi dậy tư duy hệ thống cho em trong suốt quá trình thực hiện đồ án vừa qua.

Nhờ sự hướng dẫn nhiệt tình của thầy, em đã có cơ hội tiếp cận và hiểu sâu hơn về mô hình CNN – một lĩnh vực không chỉ mang tính học thuật mà còn có ứng dụng vô cùng thiết thực trong thực tế.

Mặc dù đồ án có thể chưa hoàn hảo, nhưng em mong đây sẽ là dấu mốc giúp em rèn luyện tư duy phân tích và sẵn sàng cho những đồ án tiếp theo.

Một lần nữa, em xin chân thành cảm ơn thầy và kính chúc thầy luôn mạnh khỏe, công tác tốt, và tiếp tục truyền cảm hứng cho các thế hệ sinh viên tiếp theo.

## MỤC LỤC

<b>Trang nhận xét của giảng viên hướng dẫn.....</b>	<b>i</b>
<b>Trang nhận xét của thành viên hội đồng .....</b>	<b>ii</b>
<b>LỜI CẢM ƠN.....</b>	<b>iii</b>
<b>MỤC LỤC.....</b>	<b>iv</b>
<b>DANH MỤC HÌNH ẢNH.....</b>	<b>vii</b>
<b>DANH MỤC BẢNG BIỂU.....</b>	<b>viii</b>
<b>Tóm tắt đồ án.....</b>	<b>ix</b>
<b>MỞ ĐẦU .....</b>	<b>1</b>
Lý do chọn đề tài.....	1
Mục đích nghiên cứu .....	1
Đối tượng nghiên cứu.....	2
Phương pháp nghiên cứu .....	2
Phạm vi nghiên cứu.....	3
<b>CHƯƠNG 1. TỔNG QUAN VỀ NGHIÊN CỨU .....</b>	<b>4</b>
1.1 Giới thiệu về học máy (Machine Learning).....	4
1.1.1 Khái niệm.....	4
1.1.2 Nguyên tác hoạt động.....	4
1.1.3 Ứng dụng của học máy trong thực tế .....	5
1.2 Giới thiệu về học sâu (Deep Learning) .....	5
1.1.1 Khái niệm.....	6
1.2.2 Nguyên tác hoạt động.....	7
1.1.2 Ứng dụng thực tế.....	8
1.2 Ứng dụng Machine Learning và Deep Learning nhận dạng ảnh.....	8
<b>CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT .....</b>	<b>9</b>

2.1 Giới thiệu về mạng nơ-ron .....	9
2.1.1 Khái niệm.....	9
2.1.2 Nguyên tắc hoạt động .....	9
2.1.3 Kiến trúc của mạng nơ-ron .....	11
2.2 Giới thiệu về nơ-ron tích chập (Convolutional Neural Network - CNN) .....	12
2.2.1 Nguyên tắc hoạt động .....	12
2.2.2 Ứng dụng mạng nơ-ron tích chập trong thực tế.....	14
2.2.3 Ưu và nhược điểm của mạng nơ-ron tích chập.....	15
<b>CHƯƠNG 3. THỰC HIỆN HÓA NGHIÊN CỨU .....</b>	<b>17</b>
3.1 Mô tả mô hình .....	17
3.2 Đặc tả yêu cầu .....	17
3.3 Ngôn ngữ lập trình Python.....	18
3.4 Các thư viện sử dụng trong mô hình .....	18
3.5 Thu thập dữ liệu .....	19
3.6 Xử lý tiền dữ liệu.....	20
3.6.1 Tổ chức và chia tập dữ liệu.....	20
3.6.2 Chuẩn hóa dữ liệu ảnh .....	21
3.6.3 Thay đổi kích thước ảnh .....	22
3.6.4 Gán nhãn dữ liệu .....	22
3.7 Xây dựng mô hình .....	23
3.8 Huấn luyện mô hình .....	24
<b>CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU .....</b>	<b>26</b>
4.1 Thực nghiệm với tập ảnh .....	26
4.1.1 Thực nghiệm với ảnh test .....	26
4.1.2 Thực nghiệm với ảnh train.....	29
4.2 Đánh giá tổng thể mô hình.....	31

---

4.2.1 Kết quả huấn luyện mô hình .....	32
4.2.2 Kết quả trên tập kiểm tra (Test set) .....	32
<b>CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>32</b>
5.1 Kết luận chung của đề tài.....	32
5.2 Các kết quả nổi bật đạt được.....	33
5.3 Ý nghĩa và giá trị đề tài.....	33
5.3.1 Ý nghĩa học thuật .....	33
5.3.2 Ý nghĩa thực tiễn .....	33
5.4 Hạn chế của đề tài.....	34
5.5 Hướng phát triển trong tương lai.....	34
5.6 Kết luận cuối cùng.....	34
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>36</b>



**DANH MỤC HÌNH ẢNH**

Hình 1. 1 Minh họa học máy .....	4
Hình 1. 2 Hình minh họa giữa Machine Learning và Deep Learning .....	6
Hình 1. 3 Minh họa mô hình CNN.....	7
Hình 2. 1 Quá trình hoạt động của mạng nơ-ron .....	10
Hình 2. 2 Kiến trúc mạng nơ-ron.....	12
Hình 2. 3 Mô hình CNN đơn giản.....	14
Hình 3. 1 Minh họa Python.....	18
Hình 3. 2 Minh họa ảnh mèo .....	20
Hình 3. 3 Minh họa ảnh chó .....	20
Hình 3. 4 Minh họa ảnh voi .....	20
Hình 3. 5 Minh họa ảnh tiger.....	20
Hình 4. 1 Kết quả ảnh test của ảnh cat trên tập test .....	26
Hình 4. 2 Kết quả ảnh test của ảnh dog trên tập test.....	27
Hình 4. 3 Kết quả ảnh test của ảnh elephant trên tập test .....	27
Hình 4. 4 Kết quả test ảnh tiger trên tập test .....	28
Hình 4. 5 Kết quả test ảnh cat trên tập train .....	29
Hình 4. 6 Kết quả test ảnh dog trên tập train.....	30
Hình 4. 7 Kết quả test ảnh elephant trên tập train.....	31
Hình 4. 8 Kết quả test ảnh tiger trên tập train.....	31

**DANH MỤC BẢNG BIỂU**

Bảng 4.1 Thống kê 4 loài động vật .....	28
Bảng 4.2 Thống kê test 4 loài trên tập train.....	31

## TÓM TẮT ĐỒ ÁN

Trong bối cảnh trí tuệ nhân tạo và học sâu (Deep Learning) ngày càng phát triển mạnh mẽ, các bài toán xử lý ảnh và nhận dạng đối tượng đang được ứng dụng rộng rãi trong nhiều lĩnh vực như sinh học, nông nghiệp, giáo dục và giám sát tự động. Trong đó, bài toán phân loại động vật dựa trên hình ảnh đóng vai trò quan trọng trong việc hỗ trợ nghiên cứu, quản lý và bảo tồn các loài động vật.

Đề tài “Xây dựng mô hình CNN cho bài toán phân loại động vật” được thực hiện nhằm nghiên cứu và ứng dụng mạng nơ-ron tích chập (Convolutional Neural Network – CNN) để tự động nhận diện và phân loại các loài động vật từ hình ảnh đầu vào. Mục tiêu của đề tài là xây dựng một mô hình học sâu có khả năng học đặc trưng ảnh hiệu quả, đạt độ chính xác cao và có thể mở rộng cho các bài toán thực tế.

Hướng tiếp cận của đề tài tập trung vào việc sử dụng ngôn ngữ Python kết hợp với các thư viện học sâu phổ biến như TensorFlow, Keras, NumPy, Matplotlib. Dữ liệu hình ảnh được thu thập từ các tập dữ liệu công khai, sau đó tiến hành tiền xử lý như thay đổi kích thước ảnh, chuẩn hóa dữ liệu và tăng cường dữ liệu (data augmentation). Mô hình CNN được thiết kế gồm các lớp Convolution, Pooling, Flatten và Dense để trích xuất đặc trưng và thực hiện phân loại.

Kết quả đạt được của đề tài là một mô hình CNN có khả năng phân loại chính xác các loài động vật trong tập dữ liệu thử nghiệm, thể hiện hiệu quả rõ rệt so với các phương pháp phân loại truyền thống. Mô hình hoạt động ổn định, dễ huấn luyện và đánh giá trên môi trường Python. Ngoài ra, hệ thống có tiềm năng mở rộng để áp dụng vào các ứng dụng thực tế như nhận diện động vật hoang dã, hỗ trợ học tập trực quan, hoặc tích hợp vào các hệ thống thông minh trong tương lai.

## MỞ ĐẦU

### Lý do chọn đề tài

Trong thời đại công nghệ số hiện nay, trí tuệ nhân tạo (Artificial Intelligence) và học sâu (Deep Learning) đang phát triển mạnh mẽ và được ứng dụng rộng rãi trong nhiều lĩnh vực như nhận dạng khuôn mặt, xử lý ảnh, y tế, nông nghiệp và sinh học. Đặc biệt, các bài toán phân loại hình ảnh ngày càng nhận được nhiều sự quan tâm do khả năng tự động hóa và độ chính xác cao.

Trong thực tế, việc nhận diện và phân loại các loài động vật bằng phương pháp thủ công tốn nhiều thời gian, công sức và dễ xảy ra sai sót, nhất là khi số lượng dữ liệu hình ảnh ngày càng lớn. Trong khi đó, các mô hình học sâu, tiêu biểu là mạng nơ-ron tích chập (Convolutional Neural Network – CNN), đã chứng minh hiệu quả vượt trội trong việc trích xuất đặc trưng ảnh và phân loại đối tượng.

Từ thực tế đó, em lựa chọn đề tài “Xây dựng mô hình CNN cho bài toán phân loại động vật” nhằm:

Nghiên cứu và tìm hiểu nguyên lý hoạt động của mạng nơ-ron tích chập (CNN) trong xử lý và phân loại hình ảnh.

Ứng dụng các thư viện học sâu như TensorFlow, Keras và ngôn ngữ Python để xây dựng mô hình phân loại động vật từ hình ảnh.

Rèn luyện kỹ năng tiền xử lý dữ liệu, thiết kế mô hình, huấn luyện và đánh giá hiệu quả mô hình học sâu.

Từng bước tiếp cận và làm quen với các bài toán thực tế trong lĩnh vực trí tuệ nhân tạo và học máy.

Thông qua đề tài này, em mong muốn nâng cao kiến thức chuyên môn về học sâu, đồng thời tạo nền tảng cho việc nghiên cứu và phát triển các ứng dụng thông minh trong tương lai.

### Mục đích nghiên cứu

Về hệ thống: Xây dựng một mô hình mạng nơ-ron tích chập (Convolutional Neural Network – CNN) có khả năng phân loại các loài động vật dựa trên hình ảnh đầu vào. Mô hình thực hiện các chức năng cơ bản như: tiếp nhận ảnh, tiền xử lý dữ

liệu, huấn luyện, dự đoán và đánh giá độ chính xác. Quá trình xây dựng và huấn luyện mô hình được thực hiện bằng ngôn ngữ Python kết hợp với các thư viện học sâu như TensorFlow, Keras

*Về báo cáo kỹ thuật:* Xây dựng báo cáo kỹ thuật chi tiết trình bày toàn bộ quá trình thực hiện đề tài, bao gồm: phân tích bài toán phân loại động vật, mô tả tập dữ liệu, các bước tiền xử lý ảnh, thiết kế kiến trúc CNN, huấn luyện mô hình, đánh giá kết quả và phân tích ưu – nhược điểm của mô hình. Báo cáo giúp tổng hợp kiến thức đã học và làm cơ sở cho việc nghiên cứu, phát triển các ứng dụng trí tuệ nhân tạo nâng cao sau này.

Báo cáo kết quả kỹ thuật sẽ bao gồm các chương:

- Chương 1: Tổng quan về học máy
- Chương 2: Nghiên cứu lý thuyết
- Chương 3: Hiện thực hóa nghiên cứu
- Chương 4: Kết quả nghiên cứu
- Chương 5: Kết luận và hướng phát triển

### **Đối tượng nghiên cứu**

Đối tượng nghiên cứu của đề tài là mạng nơ-ron tích chập (Convolutional Neural Network – CNN) và việc ứng dụng mô hình này trong bài toán phân loại động vật dựa trên hình ảnh.

### **Phương pháp nghiên cứu**

*Phương pháp nghiên cứu lý thuyết:* Tìm hiểu các tài liệu, giáo trình và bài báo khoa học liên quan đến trí tuệ nhân tạo, học sâu và mạng nơ-ron tích chập (CNN). Qua đó nắm vững nguyên lý hoạt động của CNN, các kiến trúc cơ bản và phương pháp phân loại hình ảnh.

*Thu thập và chuẩn bị dữ liệu:* Sử dụng các tập dữ liệu hình ảnh động vật từ nguồn công khai. Thực hiện các bước tiền xử lý dữ liệu như chuẩn hóa ảnh, thay đổi kích thước, phân chia tập dữ liệu thành tập huấn luyện, tập kiểm tra và tập đánh giá.

### Phạm vi nghiên cứu

Đề tài tập trung vào bài toán phân loại động vật dựa trên hình ảnh tĩnh, không nghiên cứu xử lý video hoặc dữ liệu thời gian thực.

Mô hình được xây dựng và huấn luyện bằng mạng nơ-ron tích chập (CNN) với các kiến trúc cơ bản, chưa đi sâu vào các mô hình học sâu phức tạp như ResNet, EfficientNet hay Vision Transformer.

Tập dữ liệu sử dụng là dữ liệu hình ảnh động vật từ các nguồn công khai, với số lượng lớp

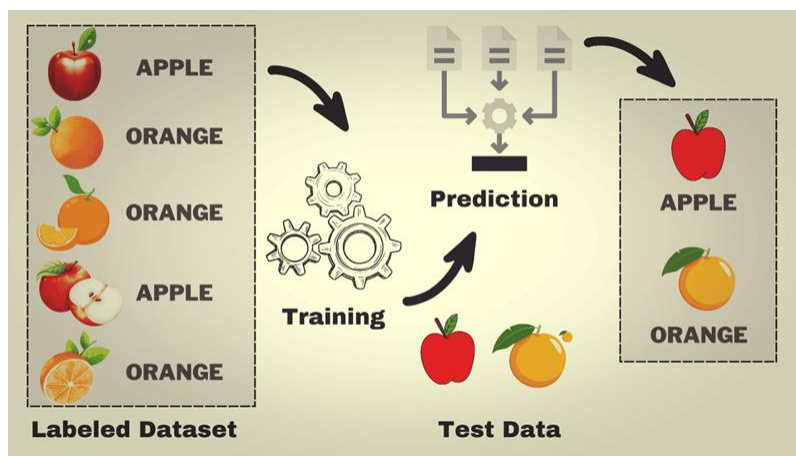
Quá trình nghiên cứu được thực hiện trên môi trường Python với các thư viện TensorFlow/Keras, huấn luyện trên máy cá nhân, không sử dụng hệ thống máy chủ hay GPU chuyên dụng. và số lượng ảnh ở mức phù hợp cho đồ án học phần.

## CHƯƠNG 1. TỔNG QUAN VỀ NGHIÊN CỨU

### 1.1 Giới thiệu về học máy (Machine Learning)

#### 1.1.1 Khái niệm

Học máy (Machine Learning) thường được viết tắt là ML, nó là một tập con của trí tuệ nhân tạo (AI) cho phép máy tính học từ dữ liệu và đưa ra dự đoán hay quyết định. Nó triển khai các thuật toán nhằm tạo điều kiện thuận lợi cho dự đoán và quyết định này. Các thuật toán được thiết kế với mục đích cải thiện hiệu suất đã trở nên ngày càng chính xác và hiệu quả hơn theo thời gian, đặc biệt khi chúng phải xử lý lượng dữ liệu lớn hơn. [1]



Hình 1. 1 Minh họa học máy

#### 1.1.2 Nguyên tắc hoạt động

##### Bước 1: Thu thập dữ liệu

– Chất lượng và số lượng của dữ liệu có thể ảnh hưởng trực tiếp đến hiệu suất mô hình. Dữ liệu được thu thập từ nhiều nguồn khác nhau như cơ sở dữ liệu, tập tin văn bản, hình ảnh, tập tin âm thanh hoặc trang web...

##### Bước 2: Tiền xử lý dữ liệu

– Quá trình tiền xử lý cải thiện chất lượng dữ liệu đảm bảo mô hình học máy có thể diễn giải dữ liệu đó một cách chính xác, cải thiện đáng kể độ chính xác của mô hình.

##### Bước 3: Chọn mô hình phù hợp

– Sau khi dữ liệu được chuẩn bị tiếp theo sẽ chọn mô hình. Có nhiều loại mô hình để lựa chọn, bao gồm hồi quy tuyến tính, cây quyết định và mạng lưới thần kinh. Lựa chọn mô hình tùy thuộc vào bản chất của dữ liệu và vấn đề đang giải quyết.

### **Bước 4: Huấn luyện mô hình**

– Quá trình huấn luyện mô hình bao gồm đưa dữ liệu vào mô hình và cho phép mô hình điều chỉnh các tham số bên trong để dự đoán kết quả đầu ra tốt hơn.

– Trong quá trình huấn luyện, điều quan trọng là tránh bị trang bị quá mức (mô hình hoạt động tốt trên dữ liệu huấn luyện nhưng kém trên dữ liệu mới) và trang bị thiếu (mô hình hoạt động kém trên cả dữ liệu huấn luyện và dữ liệu mới).

### **Bước 5: Đánh giá mô hình**

– Thử nghiệm mô hình trên dữ liệu mới mà nó chưa thấy trong quá trình đào tạo. Các số liệu phổ biến để đánh giá hiệu suất của mô hình bao gồm độ chính xác (đối với các vấn đề phân loại), độ chính xác và thu hồi (đối với các vấn đề phân loại nhị phân) và sai số bình phương trung bình (đối với các vấn đề hồi quy).

### **1.1.3 Ứng dụng của học máy trong thực tế**

Machine Learning (ML) đang được áp dụng rộng rãi trong nhiều lĩnh vực khác nhau trong thực tế. Dưới đây là một số lĩnh vực phổ biến mà ML đang được sử dụng:

- Công nghệ thông tin và Internet.
- Tài chính.
- Y tế.
- Ô tô tự hành và IoT (Internet vạn vật).
- Bán lẻ và Thương mại điện tử.
- Giáo dục.
- Quản lý chuỗi cung ứng.
- Giải trí.

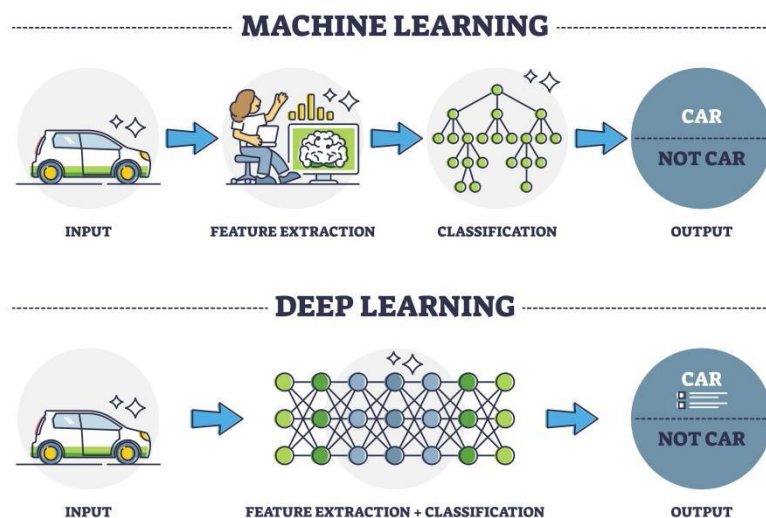
## **1.2 Giới thiệu về học sâu (Deep Learning)**



### 1.1.1 Khái niệm

Học sâu (Deep Learning) là một phần của trí tuệ nhân tạo (AI) mà các mạng nơ-ron sâu được sử dụng để học và hiểu dữ liệu phức tạp. Trong Deep Learning, các mạng nơ-ron được cấu trúc với nhiều lớp (từ đó có tên gọi "sâu"), mỗi lớp thực hiện các phép tính phức tạp để tự động rút trích các đặc trưng từ dữ liệu.[2]

Deep Learning thường được áp dụng trong các bài toán nhận dạng hình ảnh, nhận dạng giọng nói, dịch ngôn ngữ tự nhiên, và nhiều lĩnh vực khác. Đặc biệt, trong nhận dạng hình ảnh, Deep Learning đã giúp máy tính nhận dạng đối tượng và các đặc điểm phức tạp trong ảnh với hiệu suất cao hơn, gần như ngang ngửa với khả năng nhận dạng của con người.



Hình 1. 2 Hình minh họa giữa Machine Learning và Deep Learning

Tóm lại, Deep Learning là một lĩnh vực của trí tuệ nhân tạo, trong đó các mạng nơ-ron sâu được sử dụng để học và hiểu dữ liệu phức tạp, mà không cần phải xác định rõ các đặc trưng hoặc quy tắc một cách cụ thể.[3]

### 1.2.2 Nguyên tác hoạt động

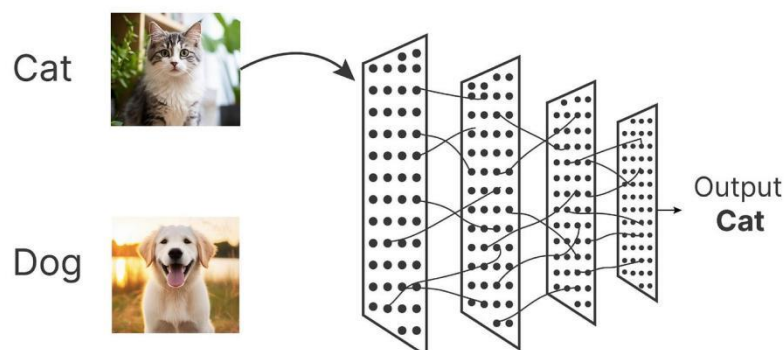
Mạng lưới thần kinh học sâu hoặc mạng lưới thần kinh nhân tạo cố gắng bắt chước bộ não con người thông qua sự kết hợp của dữ liệu đầu vào, trọng số và độ lệch. Các phần tử này phối hợp với nhau để nhận dạng, phân loại và mô tả chính xác các đối tượng trong dữ liệu.

Mạng lưới thần kinh sâu bao gồm nhiều lớp nút được kết nối với nhau, mỗi lớp được xây dựng dựa trên lớp trước đó để tinh chỉnh và tối ưu hóa dự đoán hoặc phân loại. Tiến trình tính toán này thông qua mạng được gọi là lan truyền tiến. Các lớp đầu vào và đầu ra của mạng nơ-ron sâu được gọi là các lớp hiển thị. Lớp đầu vào là nơi mô hình học sâu nhập dữ liệu để xử lý và lớp đầu ra là nơi đưa ra dự đoán hoặc phân loại cuối cùng.

Phân trên mô tả loại mạng lưới thần kinh sâu đơn giản nhất bằng những thuật ngữ đơn giản nhất. Tuy nhiên, các thuật toán học sâu cực kỳ phức tạp và có nhiều loại mạng thần kinh khác nhau để giải quyết các vấn đề hoặc bộ dữ liệu cụ thể.

Ví dụ:

– Mạng thần kinh tích chập (CNN), được sử dụng chủ yếu trong các ứng dụng phân loại hình ảnh và thị giác máy tính, có thể phát hiện các đặc điểm và mẫu trong hình ảnh, từ đó thực hiện các tác vụ như phát hiện hoặc nhận dạng đối tượng. Vào năm 2015, CNN lần đầu tiên đã đánh bại con người trong thử thách nhận dạng vật thể.



Hình 1. 3 Minh họa mô hình CNN

### 1.1.2 Ứng dụng thực tế

Deep Learning đã được áp dụng rộng rãi trong nhiều lĩnh vực khác nhau trong thực tế.[4]

- Nhận dạng hình ảnh.
- Xử lý ngôn ngữ.
- Ô tô tự hành và xe thông minh.
- Y học và chăm sóc sức khỏe.
- Tổ chức và dự báo thị trường tài chính.

### 1.2 Ứng dụng Machine Learning và Deep Learning nhận dạng ảnh

Ứng dụng quan trọng của Machine Learning và Deep Learning, nói chung, và được áp dụng rộng rãi trong nhiều lĩnh vực, bao gồm:

1. Nhận dạng khuôn mặt.
2. Nhận dạng đối tượng.
3. Phát hiện vật thể.
4. Nhận dạng biểu ngữ chữ viết tay.
5. Nhận dạng trong y học.

Đây chỉ là một số ứng dụng phổ biến của bài toán nhận dạng ảnh trong Machine Learning và Deep Learning. Đối với mỗi ứng dụng, có thể sử dụng các phương pháp và mô hình khác nhau tùy thuộc vào yêu cầu cụ thể của bài toán.

## CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT

### 2.1 Giới thiệu về mạng nơ-ron

#### 2.1.1 Khái niệm

Neural Network (Mạng nơ-ron) là một mô hình tính toán được lấy cảm hứng từ cách hoạt động của não người. Nó bao gồm một loạt các "nơ-ron" nhân tạo được tổ chức thành các lớp (layers), trong đó mỗi nơ-ron kết nối với nơ-ron trong lớp tiếp theo. Mỗi nơ-ron có thể nhận đầu vào, xử lý thông tin và gửi đầu ra đến các nơ-ron khác. [5]

Mạng nơ-ron được cấu trúc thành nhiều lớp, trong đó có ít nhất một lớp đầu vào (input layer) để nhận dữ liệu, một hoặc nhiều lớp ẩn (hidden layers) để xử lý thông tin và một lớp đầu ra (output layer) để sản xuất kết quả. Mỗi nơ-ron trong mỗi lớp có trọng số (weights) và một hàm kích hoạt (activation function), được sử dụng để tính toán đầu ra của nơ-ron dựa trên đầu vào và trọng số.

Khi được huấn luyện, mạng nơ-ron học cách điều chỉnh các trọng số của nó dựa trên dữ liệu đầu vào và các kết quả mong muốn tương ứng. Quá trình này gọi là "huấn luyện mạng nơ-ron", và thường được thực hiện bằng các thuật toán như backpropagation (truyền ngược), trong đó mạng nơ-ron điều chỉnh các trọng số của nó để giảm thiểu sai số giữa kết quả dự đoán và kết quả thực tế.

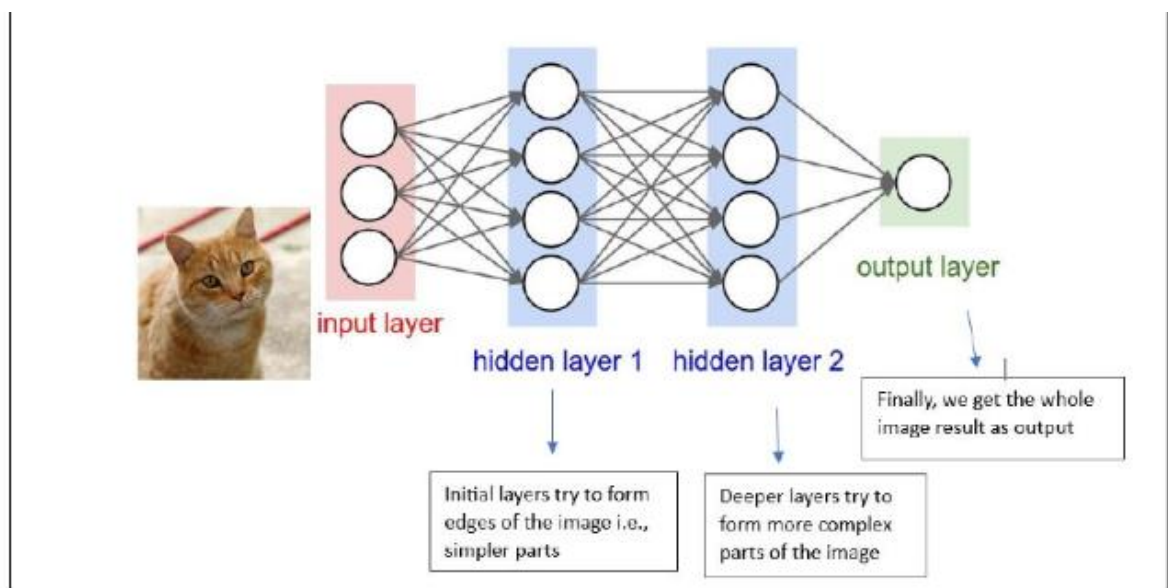
#### 2.1.2 Nguyên tắc hoạt động

Mạng nơ-ron hoạt động dựa trên một số nguyên tắc cơ bản. Dưới đây là một phác thảo về cách hoạt động của mạng nơ-ron:

1. **Nhập dữ liệu:** Đầu tiên, dữ liệu đầu vào được cung cấp cho mạng nơ-ron. Đây có thể là ảnh, văn bản, âm thanh hoặc bất kỳ loại dữ liệu nào khác.
2. **Tính toán trên nơ-ron:** Dữ liệu đầu vào được truyền qua các nơ-ron trong lớp đầu vào của mạng. Mỗi nơ-ron nhận dữ liệu, thực hiện các phép tính và truyền kết quả tới nơ-ron trong lớp tiếp theo.
3. **Tính toán trên các lớp ẩn:** Dữ liệu tiếp tục được truyền qua các lớp ẩn của mạng nơ-ron. Các lớp ẩn này thực hiện các phép tính phức tạp để xử lý thông tin và trích xuất các đặc trưng quan trọng từ dữ liệu.

4. Tính toán đầu ra: Sau khi dữ liệu đã đi qua các lớp ẩn, nó được đưa vào lớp đầu ra của mạng. Lớp đầu ra tạo ra dự đoán hoặc đầu ra dựa trên dữ liệu đã được xử lý.
5. So sánh với kết quả mong muốn: Kết quả đầu ra được so sánh với kết quả mong muốn để đo lường sự chính xác của mô hình.
6. Đánh giá và điều chỉnh: Dựa trên sự chênh lệch giữa kết quả dự đoán và kết quả mong muốn, mạng nơ-ron sẽ điều chỉnh các trọng số của nó thông qua quá trình được gọi là "huấn luyện". Quá trình này có thể được thực hiện bằng các thuật toán như backpropagation, trong đó mạng nơ-ron điều chỉnh các trọng số của nó để giảm thiểu sai số.

Dự đoán mới: Sau khi mô hình đã được huấn luyện, nó có thể được sử dụng để dự đoán các đầu vào mới mà nó chưa từng thấy trước đó. Quá trình này được lặp lại cho tới khi mạng nơ-ron đạt được hiệu suất mong muốn và có thể dự đoán đầu ra một cách chính xác.



Hình 2. 1 Quá trình hoạt động của mạng nơ-ron

### 2.1.3 Kiến trúc của mạng nơ-ron

Kiến trúc mạng nơ-ron đề cập đến cách mà các nơ-ron và các lớp của chúng được tổ chức và kết nối lại với nhau. Dưới đây là một số kiến trúc mạng nơ-ron phổ biến:

#### 1. Mạng nơ-ron một lớp (Single-Layer Perceptron):

- Đây là kiến trúc cơ bản nhất của mạng nơ-ron.
- Bao gồm một lớp đầu vào và một lớp đầu ra.
- Mỗi nơ-ron trong lớp đầu vào kết nối với tất cả các nơ-ron trong lớp đầu ra.

#### 2. Mạng nơ-ron truyền thẳng (Feedforward Neural Network - FNN).

- Là một mạng nơ-ron có cấu trúc tuyến tính, dữ liệu di chuyển theo một hướng từ lớp đầu vào đến lớp đầu ra mà không có chu trình phản hồi.
- Bao gồm một hoặc nhiều lớp ẩn giữa lớp đầu vào và lớp đầu ra.

#### 3. Mạng nơ-ron tái phân cấp (Recurrent Neural Network - RNN):

- Mạng nơ-ron có khả năng lưu trữ thông tin về trạng thái trước đó và sử dụng nó trong quá trình dự đoán đầu ra tiếp theo.
- Có thể xử lý dữ liệu dạng chuỗi hoặc dữ liệu có thứ tự, như ngôn ngữ tự nhiên, dữ liệu thời gian và dữ liệu chuỗi.
- Thích hợp cho các bài toán dự đoán chuỗi thời gian, dịch thuật máy, tạo văn bản tự động và nhiều ứng dụng khác.

#### 4. Mạng nơ-ron hồi tiếp dài (Long Short-Term Memory - LSTM):

- Một loại mạng nơ-ron tái phân cấp, được thiết kế để giải quyết vấn đề biến mất gradient trong RNN.
- Có khả năng ghi nhớ thông tin trong thời gian dài và ngắn hạn.
- Thích hợp cho các bài toán như dự đoán chuỗi thời gian, xử lý ngôn ngữ tự nhiên và nhiều ứng dụng khác đòi hỏi khả năng lưu trữ thông tin trong thời gian dài.

#### 5. Mạng nơ-ron tích chập (Convolutional Neural Network - CNN):

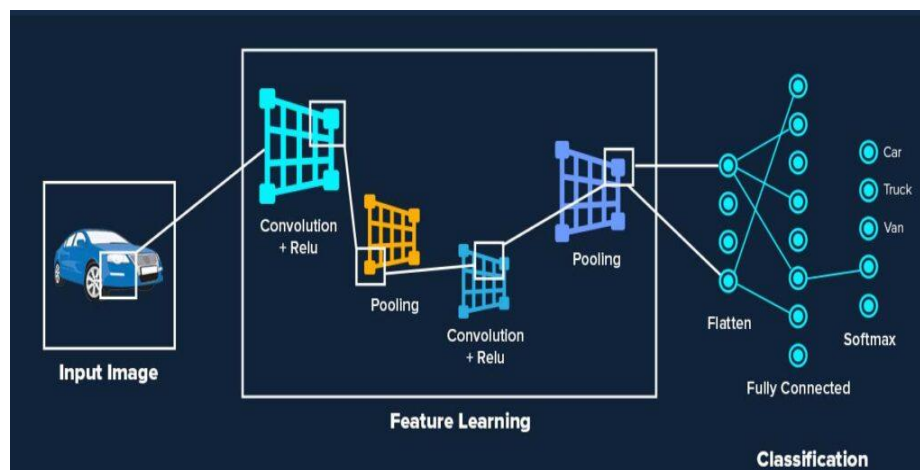
- Thiết kế đặc biệt cho việc xử lý hình ảnh và video.

- Sử dụng lớp tích chập để tự động trích xuất đặc trưng từ dữ liệu hình ảnh.
- Thích hợp cho các bài toán nhận dạng hình ảnh, phát hiện vật thể, phân loại hình ảnh và nhiều ứng dụng liên quan đến xử lý hình ảnh.

Mỗi kiến trúc mạng nơ-ron phù hợp với một loại bài toán và có ưu điểm và hạn chế riêng. Sự lựa chọn kiến trúc phù hợp sẽ phụ thuộc vào yêu cầu cụ thể của vấn đề và tính chất của dữ liệu.

### 2.2 Giới thiệu về nơ-ron tích chập (Convolutional Neural Network - CNN)

Mạng nơ-ron tích chập (CNN) là một loại mạng nơ-ron nhân tạo được thiết kế đặc biệt để xử lý dữ liệu dạng lưới như hình ảnh và video. CNN thường được sử dụng trong các ứng dụng liên quan đến thị giác máy tính và nhận dạng hình ảnh. Đặc điểm chính của CNN là sự áp dụng của các lớp tích chập, lớp gộp (pooling), và các lớp kích hoạt phi tuyến tính, giúp mô hình có khả năng học các đặc trưng cục bộ trong dữ liệu đầu vào một cách hiệu quả. Điều này giúp CNN có khả năng tự động rút trích ra các đặc trưng quan trọng từ dữ liệu ảnh một cách tự nhiên, giúp cho việc nhận dạng và phân loại trở nên hiệu quả hơn. [6]



Hình 2. 2 Kiến trúc mạng nơ-ron

#### 2.2.1 Nguyên tắc hoạt động

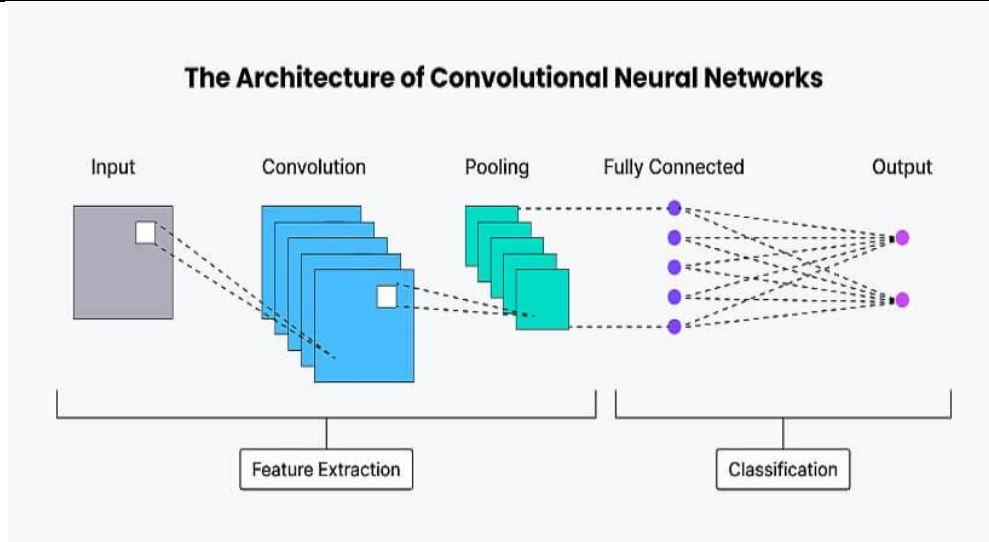
Trong thuật toán CNN, Feature (đặc trưng) thường được hiểu là các biểu diễn hoặc đặc tính quan trọng của dữ liệu hình ảnh được học bởi mô hình. Các đặc trưng này thường là những thuộc tính cụ thể của hình ảnh như cạnh, góc, hoặc các đặc điểm phức tạp hơn như mắt, mũi, miệng trong trường hợp nhận diện khuôn mặt.

Trong quá trình huấn luyện CNN, các layer tích chập của mô hình học cách trích xuất các đặc trưng từ dữ liệu hình ảnh đầu vào. Các layer sau đó có thể kết hợp các đặc trưng này để tạo ra biểu diễn ngày càng phức tạp và trừu tượng của hình ảnh. Các đặc trưng này cuối cùng được sử dụng để phân loại hoặc thực hiện các nhiệm vụ khác liên quan đến xử lý hình ảnh.

Nguyên tắc hoạt động của mạng tích chập (CNN) có thể được phân thành các bước chính như sau [7]:

- **Lớp tích chập (Convolution):** Bước này là lõi của CNN. Các bộ lọc tích chập được áp dụng trên dữ liệu đầu vào để tạo ra các đặc trưng ẩn. Mỗi bộ lọc sẽ thực hiện việc nhân chập (convolution) trên vùng nhỏ của dữ liệu đầu vào để tạo ra một feature map.
- **Hàm kích hoạt (Activation):** Hàm ReLU giúp mô hình học những đặc trưng chưa được tìm thấy trong dữ liệu và làm cho quá trình huấn luyện mô hình trở nên hiệu quả hơn.
- **Lớp gộp (Pooling):** Bước gộp giúp giảm kích thước của feature map bằng cách thực hiện các phép tổng hợp trên các vùng cục bộ của feature map. Điều này giúp giảm lượng tham số trong mô hình và làm giảm độ phức tạp của quá trình học.
- **Bước lặp lại (Repeat):** Các bước tích chập, kích hoạt và gộp có thể được lặp lại nhiều lần để tạo ra các layer ẩn sâu trong mạng.
- **Lớp kết nối đầy đủ (Fully Connected – Dense):** Sau khi các layer tích chập và gộp, các feature map được duỗi thành một vector và được đưa vào các lớp nơ-ron đầy đủ (fully connected) để thực hiện việc phân loại hoặc dự đoán.
- **Đầu ra (Output):** Cuối cùng, một lớp đầu ra được sử dụng để tạo ra dự đoán cho bài toán cụ thể, như phân loại hình ảnh thành các lớp khác nhau hoặc dự đoán các giá trị trong trường hợp của học có giám sát.





Hình 2. 3 Mô hình CNN đơn giản

### 2.2.2 Ứng dụng mạng nơ-ron tích chập trong thực tế

Mạng nơ-ron tích chập (CNN) đã có nhiều ứng dụng quan trọng và đa dạng trong thực tế, bao gồm:

- **Xử lý ngôn ngữ tự nhiên (NLP):** Trong NLP, CNN có thể được sử dụng để phân loại văn bản, đánh giá cảm xúc trong văn bản, và thậm chí là dịch máy.
- **Tự động hóa làm việc:** CNN có thể hỗ trợ trong việc tự động hóa quy trình công việc, chẳng hạn như nhận diện và phân loại tài liệu.
- **Y tế và chăm sóc sức khỏe:** CNN được sử dụng trong hình ảnh y tế để phân loại bệnh lý, nhận diện tế bào ung thư, và hỗ trợ trong quá trình chẩn đoán.
- **Tìm kiếm và gợi ý:** CNN có thể cải thiện kết quả tìm kiếm hình ảnh và cung cấp gợi ý sản phẩm dựa trên hình ảnh.
- **Xe tự hành và công nghệ ô tô:** Trong xe tự hành, CNN có vai trò quan trọng trong việc nhận diện và đánh giá môi trường xung quanh.
- **Trò chơi và giải trí:** CNN được sử dụng trong công nghiệp trò chơi để tạo đồ họa chất lượng cao và cải thiện trải nghiệm người chơi.
- **Phân loại dữ liệu:** CNN (Convolutional Neural Network) thường được sử dụng để phân loại dữ liệu phức tạp trong nghiên cứu và khoa học dữ liệu.
- **Nông nghiệp thông minh:** CNN có thể hỗ trợ trong nhận diện và theo dõi mặt đất, dự đoán mùa vụ và quản lý tình trạng nông nghiệp. Các ứng dụng

của CNN liên tục mở rộng, và nó trở thành một công nghệ quan trọng trong lĩnh vực trí tuệ nhân tạo và xử lý ảnh.

### 2.2.3 Ưu và nhược điểm của mạng nơ-ron tích chập

Dưới đây là một số ưu điểm và nhược điểm của mạng nơ-ron tích chập (CNN):

#### **Ưu điểm:**

1. Khả năng học đặc trưng cục bộ: CNN có khả năng học các đặc trưng cục bộ từ dữ liệu đầu vào, giúp nó phù hợp cho việc xử lý dữ liệu có cấu trúc dạng lưới như hình ảnh, video.
2. Chia sẻ trọng số (Weight Sharing): CNN sử dụng chia sẻ trọng số trong các bộ lọc tích chập, giúp giảm lượng tham số cần học, tăng tính hiệu quả và khả năng tổng quát hóa của mô hình.
3. Thu gọn dữ liệu (Downsampling): Các bước gộp (pooling) trong CNN giúp giảm kích thước của dữ liệu, giảm độ phức tạp của mô hình và nguy cơ overfitting.
4. Hiệu suất cao trong việc nhận dạng hình ảnh: CNN đã chứng minh hiệu suất cao trong các nhiệm vụ nhận dạng hình ảnh, như phân loại, nhận dạng đối tượng, và phát hiện vật thể.
5. Tính tự động hóa cao: Khi được huấn luyện đúng cách, CNN có khả năng tự động học và rút trích các đặc trưng quan trọng từ dữ liệu đầu vào mà không cần sự can thiệp nhiều từ con người.

#### **Nhược điểm:**

1. Cần nhiều dữ liệu huấn luyện: CNN cần một lượng lớn dữ liệu huấn luyện để đạt được hiệu suất tốt, đặc biệt là khi xử lý các nhiệm vụ phức tạp.
2. Yêu cầu tính toán cao: Mạng CNN thường yêu cầu tính toán cao, đặc biệt là với các mô hình sâu và kích thước lớn.
3. Khó hiểu và khó diễn giải: Mặc dù CNN có thể đạt được hiệu suất cao, nhưng các quyết định của nó thường khó hiểu và khó diễn giải, đặc biệt là đối với các bộ lọc tích chập ẩn.

4. Dễ bị overfitting: CNN có nguy cơ cao bị overfitting, đặc biệt là khi sử dụng các mô hình phức tạp và không có đủ dữ liệu huấn luyện.
5. Không đối xứng và không thích ứng: Mặc dù CNN có thể tốt trong việc nhận dạng đối tượng trong hình ảnh, nhưng nó không đối xứng và không thích ứng, tức là việc dịch chuyển hoặc xoay hình ảnh có thể ảnh hưởng đến hiệu suất của mô hình.

### **3.1 Mô tả mô hình**

Đề tài “Xây dựng mô hình CNN cho phân loại động vật” nhằm mục đích giải quyết bài toán nhận diện và phân loại các loài động vật dựa trên hình ảnh thông qua việc ứng dụng mạng nơ-ron tích chập (Convolutional Neural Network – CNN). Mô hình được xây dựng nhằm tự động trích xuất đặc trưng từ ảnh đầu vào và đưa ra kết quả phân loại một cách chính xác và hiệu quả.

Mô hình CNN cho phép tiếp nhận ảnh động vật, thực hiện các bước tiền xử lý, sau đó tiến hành huấn luyện và dự đoán loài động vật tương ứng. Hệ thống hỗ trợ phân chia dữ liệu thành tập huấn luyện và tập kiểm tra, đánh giá kết quả thông qua các chỉ số như độ chính xác (accuracy) và hàm mất mát (loss). Ngoài ra, mô hình còn lưu lại lịch sử huấn luyện, giúp theo dõi quá trình học của mô hình và so sánh hiệu quả giữa các lần huấn luyện khi cần thiết.

### **3.2 Đặc tả yêu cầu**

#### **1. Yêu cầu chức năng**

Mô hình CNN phân loại động vật cần đáp ứng các yêu cầu chức năng sau: Nhận đầu vào là hình ảnh động vật có định dạng phổ biến như JPG, PNG.

Thực hiện tiền xử lý ảnh bao gồm thay đổi kích thước, chuẩn hóa giá trị pixel và chuyển đổi dữ liệu phù hợp với mô hình. Tự động trích xuất đặc trưng ảnh thông qua các lớp tích chập (Convolution) và gộp (Pooling). Thực hiện phân loại ảnh và đưa ra nhãn động vật tương ứng. Hỗ trợ huấn luyện, đánh giá và dự đoán trên dữ liệu mới. Lưu và tải lại mô hình đã huấn luyện để tái sử dụng.

#### **2. Yêu cầu dữ liệu**

Dữ liệu đầu vào là tập ảnh động vật được phân chia theo từng lớp (loài).

Dữ liệu được chia thành tập huấn luyện và tập kiểm tra theo tỷ lệ xác định.

Ảnh đầu vào được chuẩn hóa về kích thước 128×128 pixel, 3 kênh màu (RGB).

Nhãn dữ liệu được biểu diễn dưới dạng one-hot encoding.

#### **3. Yêu cầu môi trường**

Ngôn ngữ lập trình: Python.

Thư viện sử dụng: TensorFlow, Keras, NumPy, OpenCV, Matplotlib.

Hệ điều hành: Windows hoặc Linux.

Môi trường chạy: CPU (không yêu cầu GPU chuyên dụng).

### 3.3 Ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình mức cao, có cú pháp đơn giản và dễ đọc, giúp nhà phát triển viết mã một cách dễ dàng và nhanh chóng.

Thư viện Python: Thư viện Python phổ biến như Matplotlib, Pandas, Numpy, Requests, OpenCV-Python, Keras.



Hình 3. 1 Minh họa Python

### 3.4 Các thư viện sử dụng trong mô hình

Tên thư viện	Chức năng
Os	Dùng để thao tác với hệ thống thư mục và tệp, như đọc danh sách thư mục, tạo thư mục mới
Shutil	Hỗ trợ sao chép tệp ảnh từ thư mục dữ liệu gốc sang thư mục train và test.

## Mô Hình CNN Phân Loại Động Vật

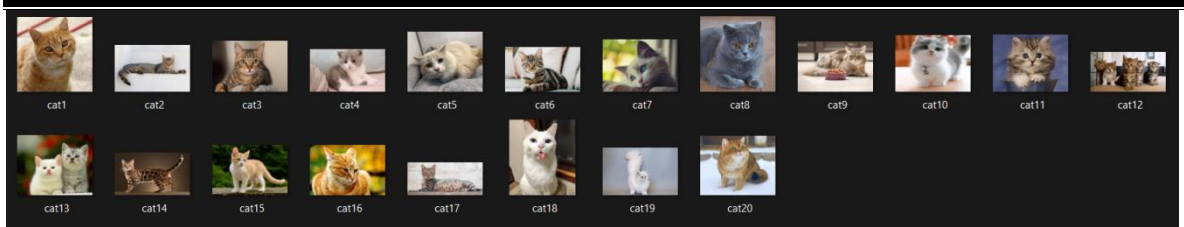
Random	Dùng để xáo trộn (shuffle) dữ liệu ảnh trước khi chia tập huấn luyện và kiểm tra, đảm bảo tính ngẫu nhiên.
Tensorflow	Thư viện chính để xây dựng, huấn luyện và lưu mô hình CNN.
Tensorflow.keras.models (Sequential)	Dùng để xây dựng mô hình mạng nơ-ron theo kiến trúc tuần tự.
Tensorflow.keras.layers (Conv2D, MaxPooling2D, Flatten, Dense)	Cung cấp các lớp cần thiết để xây dựng kiến trúc CNN.
Tensorflow.keras.preprocessing.image (ImageDataGenerator)	Dùng để đọc ảnh từ thư mục, tiền xử lý và tạo dữ liệu huấn luyện cho mô hình.
cv2 (OpenCV)	Dùng để đọc ảnh, xử lý ảnh và chuyển đổi không gian màu.
Numpy	Hỗ trợ xử lý mảng số, chuẩn hóa dữ liệu và chuyển đổi định dạng đầu vào cho mô hình
Matplotlib.pyplot	Dùng để hiển thị hình ảnh và kết quả dự đoán trực quan.

### 3.5 Thu thập dữ liệu

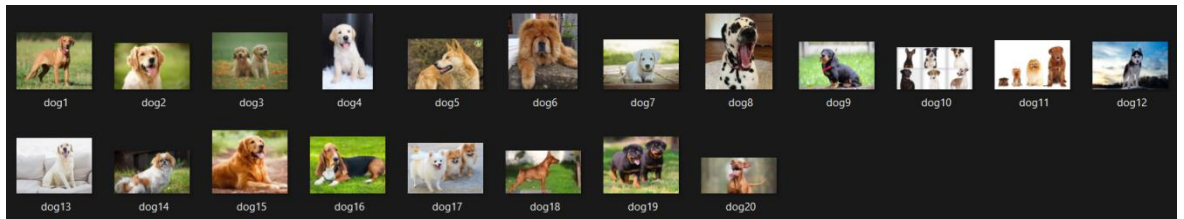
Bộ dữ liệu “Dog-Cat-Tiger-Elephant” được lấy ngẫu nhiên trên Internet bao gồm 80 bức ảnh chia thành 2 tập là tập huấn luyện và tập kiểm thử, trong đó 20 bức ảnh là ảnh con mèo, 20 bức ảnh là ảnh con chó, 20 bức ảnh là ảnh con voi, 20 bức ảnh là ảnh con hổ. Ảnh con mèo được gán nhãn thông qua đặt tên file là “cat.x” đặt trong thư mục cat, và ảnh con chó được gán nhãn thông qua đặt tên file là “dog.x” đặt trong thư mục dogs, và ảnh con voi được gán nhãn thông qua đặt tên file là “voi.x”, và ảnh con hổ được gán nhãn thông qua đặt tên file là “tiger.x”

Hình sau minh họa một số ảnh được gán nhãn:

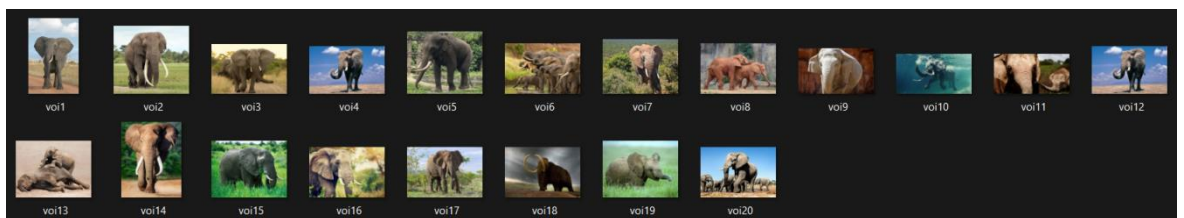
## Mô Hình CNN Phân Loại Động Vật



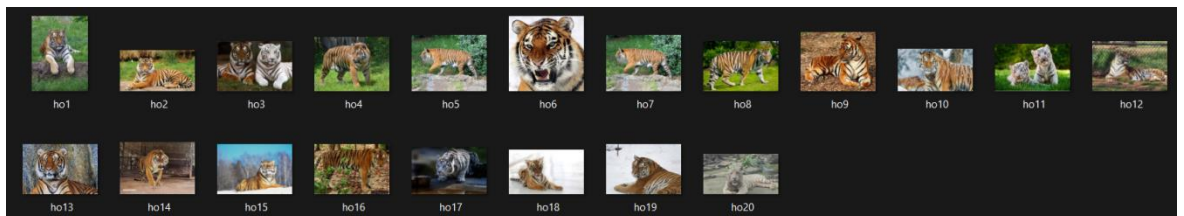
Hình 3. 2 Minh họa ảnh mèo



Hình 3. 3 Minh họa ảnh chó



Hình 3. 4 Minh họa ảnh voi



Hình 3. 5 Minh họa ảnh tiger

Tập dữ liệu huấn luyện được tổ chức riêng trong 2 thư mục: train(tập huấn luyện) và thư mục test. Tập huấn luyện là tập dữ liệu được sử dụng để huấn luyện mô hình. Tập kiểm thử bao gồm các ảnh không trùng với ảnh trong tập huấn luyện và được tổ chức lưu trữ trong thư mục test.

### 3.6 Xử lý tiền dữ liệu

#### 3.6.1 Tổ chức và chia tập dữ liệu

Tập dữ liệu hình ảnh động vật ban đầu được lưu trữ trong thư mục gốc, trong đó mỗi thư mục con đại diện cho một loài động vật. Dữ liệu được chia thành hai tập:

Tập huấn luyện (train): dùng để huấn luyện mô hình CNN.

Tập kiểm tra (test): dùng để đánh giá hiệu quả của mô hình. Việc chia dữ liệu được thực hiện tự động bằng chương trình `split_dataset.py` với tỷ lệ 70% cho tập huấn luyện và 30% cho tập kiểm tra, đảm bảo dữ liệu được xáo trộn ngẫu nhiên trước khi chia.

```
# ===== CẤU HÌNH =====
RAW_DIR = "raw_dataset"
OUTPUT_DIR = "animal_dataset"
TRAIN_RATIO = 0.7 # 70% train - 30% test

random.seed(42)

# ===== TẠO THƯ MỤC =====
for folder in ["train", "test"]:
    for cls in os.listdir(RAW_DIR):
        os.makedirs(os.path.join(OUTPUT_DIR, folder, cls), exist_ok=True)
```

Lấy danh sách tất cả ảnh trong từng lớp động vật. Xáo trộn danh sách ảnh để đảm bảo tính ngẫu nhiên khi chia dữ liệu.

```
# ===== CHIA ẢNH =====
for cls in os.listdir(RAW_DIR):
    images = os.listdir(os.path.join(RAW_DIR, cls))
    random.shuffle(images)
```

### 3.6.2 Chuẩn hóa dữ liệu ảnh

Chuẩn hóa dữ liệu ảnh Các ảnh trong tập dữ liệu có kích thước và giá trị pixel khác nhau. Do đó, ảnh được chuẩn hóa giá trị pixel về khoảng  $[0,1]$  bằng cách chia cho 255. Việc chuẩn hóa giúp mô hình hội tụ nhanh hơn và ổn định hơn trong quá trình huấn luyện.

Sử dụng `ImageDataGenerator` của Keras (trong `train_cnn.py`):

```
train_gen = ImageDataGenerator(rescale=1./255)
test_gen = ImageDataGenerator(rescale=1./255)
```

Dòng `rescale=1./255` sẽ tự động chia toàn bộ giá trị pixel (0–255) cho 255 trước khi đưa ảnh vào mô hình CNN.



Chuẩn hóa khi dự đoán ảnh mới (trong predict.py):

```
# Đọc ảnh
img_path = "animal_dataset/test/cat/10.jpg"
img = cv2.imread(img_path)
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

# Resize + chuẩn hóa
img_resized = cv2.resize(img_rgb, (128, 128))
img_norm = img_resized / 255.0
img_input = np.expand_dims(img_norm, axis=0)
```

### 3.6.3 Thay đổi kích thước ảnh

Resize ảnh khi huấn luyện (trong train\_cnn.py)

Resize được thực hiện tự động thông qua ImageDataGenerator:

```
train_data = train_gen.flow_from_directory(
    TRAIN_DIR,
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode="categorical"
```

### 3.6.4 Gán nhãn dữ liệu

Dữ liệu ảnh được gán nhãn tự động dựa trên tên thư mục chứa ảnh. Các nhãn được mã hóa dưới dạng one-hot encoding thông qua ImageDataGenerator của Keras, phù hợp với hàm mất mát categorical\_crossentropy

Mỗi thư mục con trong TRAIN\_DIR được xem là một lớp (label)

Tên thư mục chính là tên nhãn của ảnh.

- animal\_dataset/train/cat
- animal\_dataset/train/dog
- animal\_dataset/train/elephant
- animal\_dataset/train/tiger

Kiểm tra thứ tự nhãn dán print(train\_data.class\_indices)

cat  $\rightarrow [1, 0, 0, 0] \rightarrow 0$

dog  $\rightarrow [0, 1, 0, 0] \rightarrow 1$

elephant  $\rightarrow [0, 0, 1, 0] \rightarrow 2$

tiger  $\rightarrow [0, 0, 0, 1] \rightarrow 3$

### 3.7 Xây dựng mô hình

Mô hình Convolutional Neural Network (CNN) được xây dựng theo kiến trúc tuần tự (Sequential), gồm các tầng tích chập, tầng gộp, tầng kết nối đầy đủ và tầng đầu ra. Kiến trúc mô hình được thiết kế nhằm trích xuất đặc trưng từ ảnh đầu vào và thực hiện phân loại động vật một cách hiệu quả.

**1. Tầng đầu vào (Input Layer):** Ảnh đầu vào được chuẩn hóa và resize về kích thước  $128 \times 128$  pixel. Ảnh màu RGB gồm 3 kênh màu.

**2. Khối tích chập thứ nhất (Convolution Block 1):** Conv2D (32 filters, kernel  $3 \times 3$ ) Trích xuất các đặc trưng mức thấp như cạnh và đường nét. Sử dụng hàm kích hoạt ReLU để tăng khả năng học phi tuyến. MaxPooling2D ( $2 \times 2$ ) Giảm kích thước không gian đặc trưng. Giữ lại thông tin quan trọng nhất.

**3. Khối tích chập thứ hai (Convolution Block 2):** Tăng số lượng bộ lọc lên 64 giúp mô hình học được các đặc trưng phức tạp hơn như hình dạng, kết cấu. Sự khác biệt giữa các loài động vật.

**4. Tầng Flatten:** Chuyển tensor đặc trưng 3D thành vector 1D, chuẩn bị dữ liệu cho các tầng fully connected.

**5. Tầng kết nối đầy đủ (Fully Connected Layer):** Tăng khả năng phân loại mô hình.

**6. Tầng đầu ra (Output layer):** Số neuron tương ứng với số lớp động vật. Hàm Softmax trả về xác suất cho từng lớp. Phù hợp với bài toán phân loại ảnh đa lớp.

Thứ tự	Tầng	Thông Số
1	Input	$128 \times 128 \times 3$
2	Conv2D + ReLU	32 filters, $3 \times 3$

## Mô Hình CNN Phân Loại Động Vật

3	MaxPooling	2 x 2
4	Conv2D + ReLU	64 filters, 3 x 3
5	MaxPooling	2 x 2
6	Flatten	
7	Dense + ReLU	64 neurons
8	Dense + Softmax	Số lớp

### 3.8 Huấn luyện mô hình

Huấn luyện mô hình CNN phân loại động vật. Sau khi xây dựng kiến trúc mô hình và chuẩn bị dữ liệu, mô hình CNN được tiến hành huấn luyện nhằm học các đặc trưng từ tập dữ liệu ảnh và tối ưu khả năng phân loại các loài động vật.

#### 1. Biên dịch mô hình:

```
✓ model.compile(  
    optimizer="adam",  
    loss="categorical_crossentropy",  
    metrics=["accuracy"]  
)
```

Optimizer – Adam: Kết hợp ưu điểm của SGD và RMSProp, giúp mô hình hội tụ nhanh và ổn định.

Loss function – categorical\_crossentropy: Phù hợp với bài toán phân loại đa lớp, yêu cầu nhãn dữ liệu dạng one-hot encoding.

Accuracy: Đánh giá tỷ lệ dự đoán đúng của mô hình.

#### 2. Quá trình huấn luyện mô hình:

```
✓ model.fit(  
    train_data,  
    epochs=EPOCHS,  
    validation_data=test_data  
)
```

- Mô hình được huấn luyện trên tập train\_data.
- Sau mỗi epoch, mô hình được đánh giá trên tập test\_data.
- Giúp theo dõi: Độ chính xác (accuracy), hàm mất mát (loss), hiện tượng overfitting / underfitting.

### 3. Lưu mô hình huấn luyện:

```
model.save("animal_cnn.h5")  
print("✅ LƯU MODEL XONG")
```

Mô hình sau huấn luyện được lưu dưới định dạng .h5, có thể sử dụng lại cho :dự đoán ảnh mới, triển khai ứng dụng thực tế huấn luyện tiếp hoặc mở rộng mô hình.

### 4. Thực hiện dự đoán

```
# Dự đoán  
pred = model.predict(img_input)  
index = np.argmax(pred)  
label = class_names[index]
```

Mô hình sẽ trả về xác suất cho từng lớp. Lớp có xác suất cao nhất được chọn làm kết quả dự đoán.

### 5. Hiển thị ảnh và kết quả dự đoán

```
# HIỂN THỊ ẢNH + KẾT QUẢ  
plt.imshow(img_rgb)  
plt.title(f"Dự đoán: {label}")  
plt.axis("off")  
plt.show()
```

Hiển thị ảnh gốc và hiển thị nhãn động vật được dự đoán.

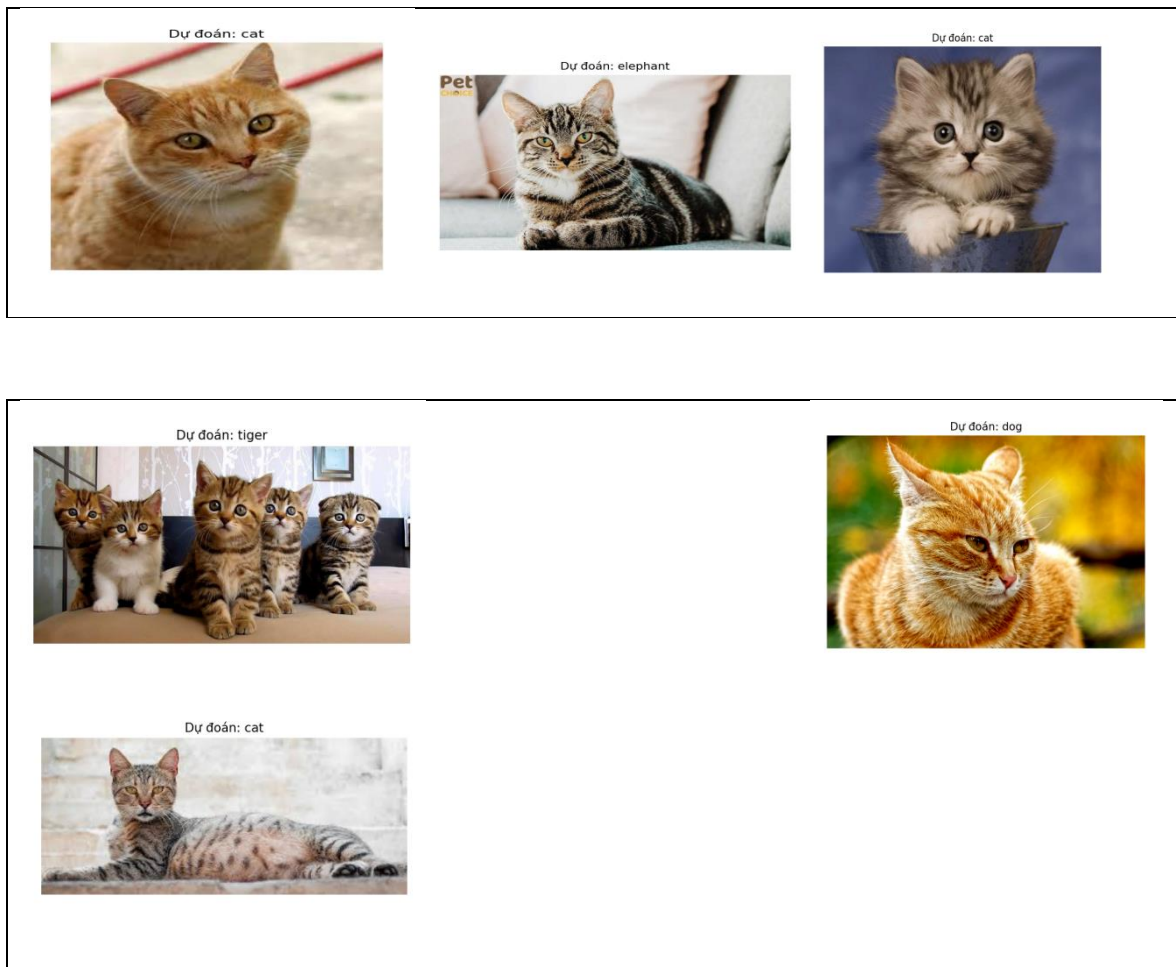
## CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU

### 4.1 Thực nghiệm với tập ảnh

#### 4.1.1 Thực nghiệm với ảnh test

Dữ liệu test được lấy trong thư mục test, đây là những ảnh không nằm trong tập ảnh huấn luyện. Dữ liệu này đã được tiền xử lý và lưu trong mảng test.

Thực nghiệm sẽ tiến hành test lấy 6 ảnh của thư mục cat kết quả như sau:



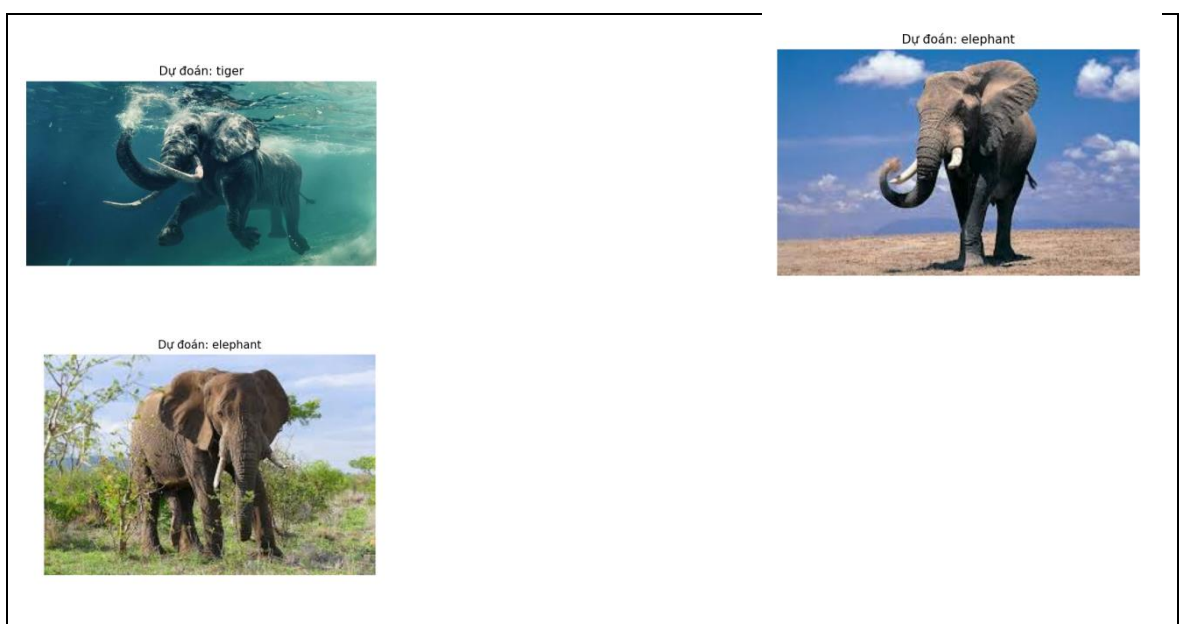
Hình 4. 1 Kết quả ảnh test của ảnh cat trên tập test

Thực nghiệm sẽ tiến hành test lấy 6 ảnh của thư mục dog kết quả như sau:



Hình 4. 2 Kết quả ảnh test của ảnh dog trên tập test

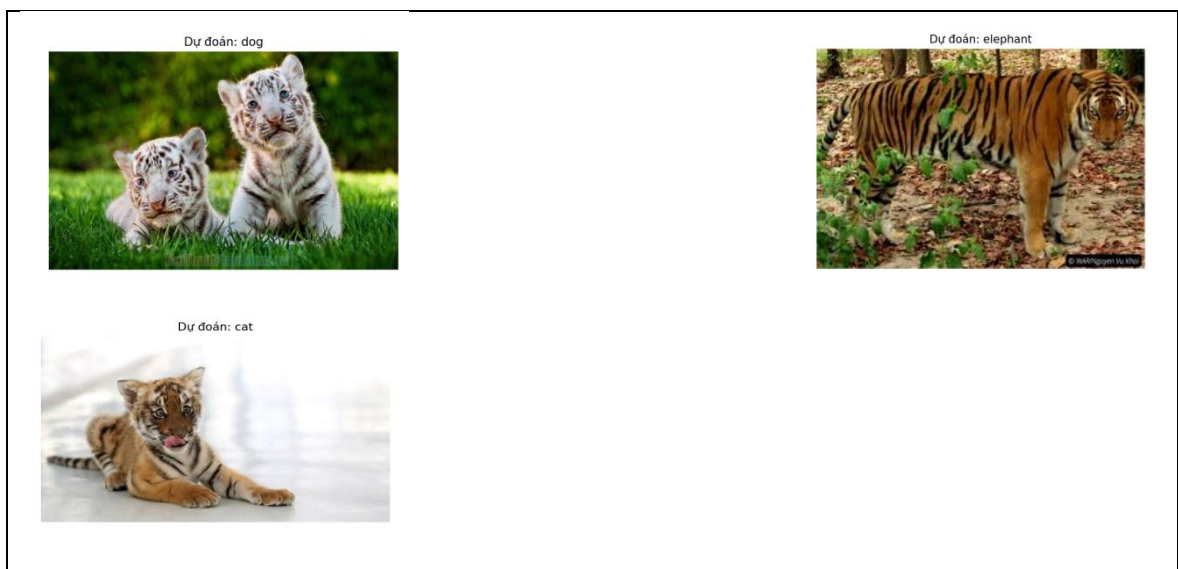
Thực nghiệm sẽ tiến hành test lấy 6 ảnh của thư mục elephant kết quả như sau:



Hình 4. 3 Kết quả ảnh test của ảnh elephant trên tập test



Thực nghiệm sẽ tiến hành test lấy 6 ảnh của thư mục tiger kết quả như sau:



Hình 4. 4 Kết quả test ảnh tiger trên tập test

Bảng thống kê kết quả test:

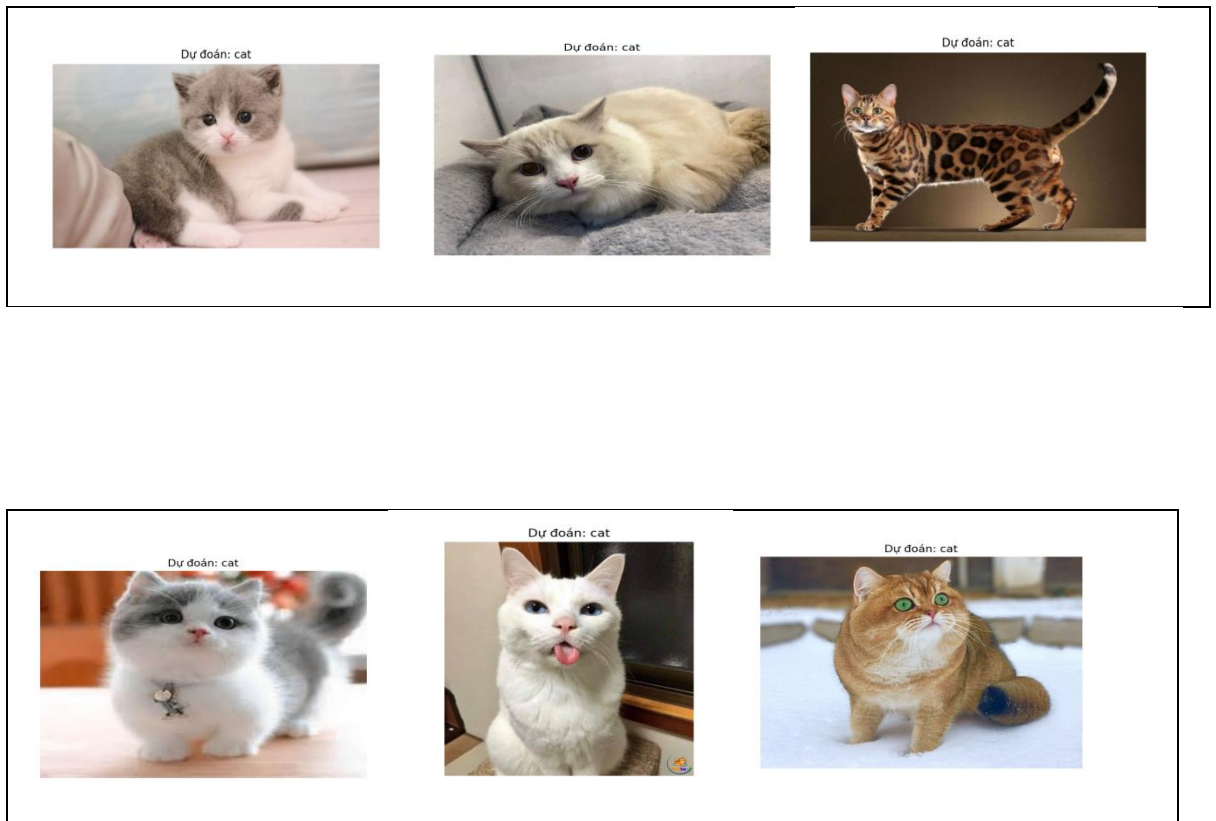
Tên loài	Số ảnh dự đoán đúng	Số ảnh dự đoán sai	Tỷ lệ đúng
Cat	3	3	50%
Dog	4	2	66%
Elephant	4	2	66%
Tiger	2	4	33%

Bảng 4.1 Thống kê 4 loài động vật

Trong bảng độ chính xác cao nhất là 66% và độ chính xác thấp nhất là 33%

### 4.4.2 Thực nhiệm với ảnh train

Thử nghiệm này được thực hiện trên các ảnh nằm trong tập huấn luyện. Kịch bản test cũng được thực hiện mỗi test lấy ngẫu nhiên 6 ảnh để dự đoán. Kết quả như ảnh cat như sau:

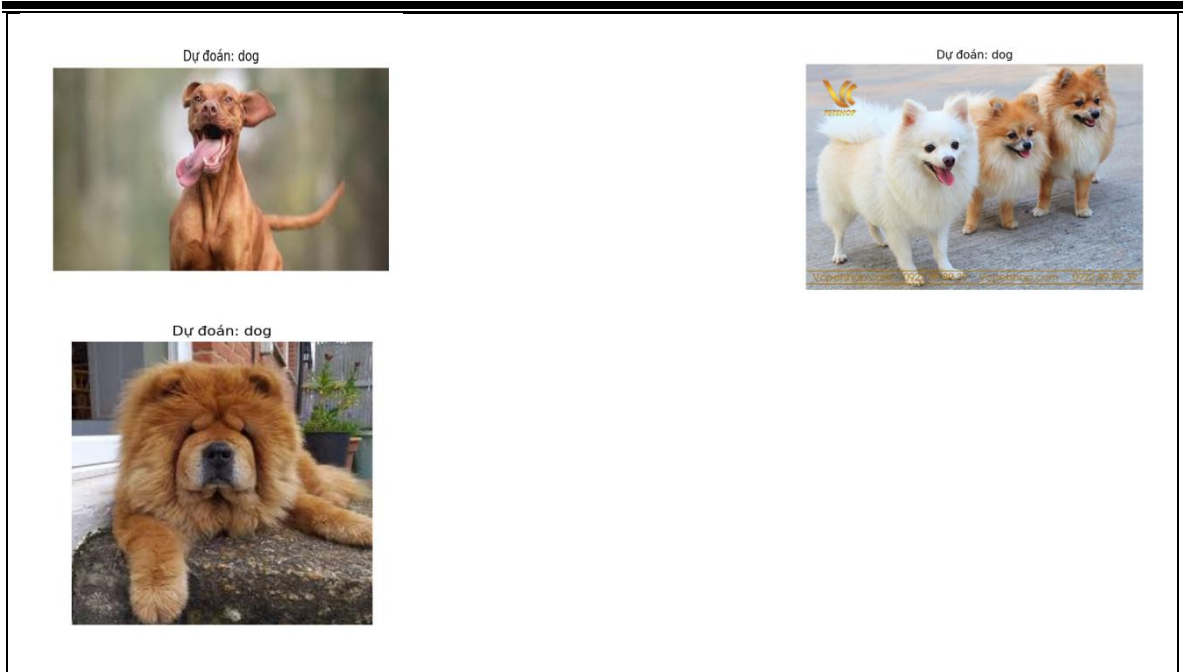


Hình 4. 5 Kết quả test ảnh cat trên tập train

Kết quả như ảnh dog như sau:

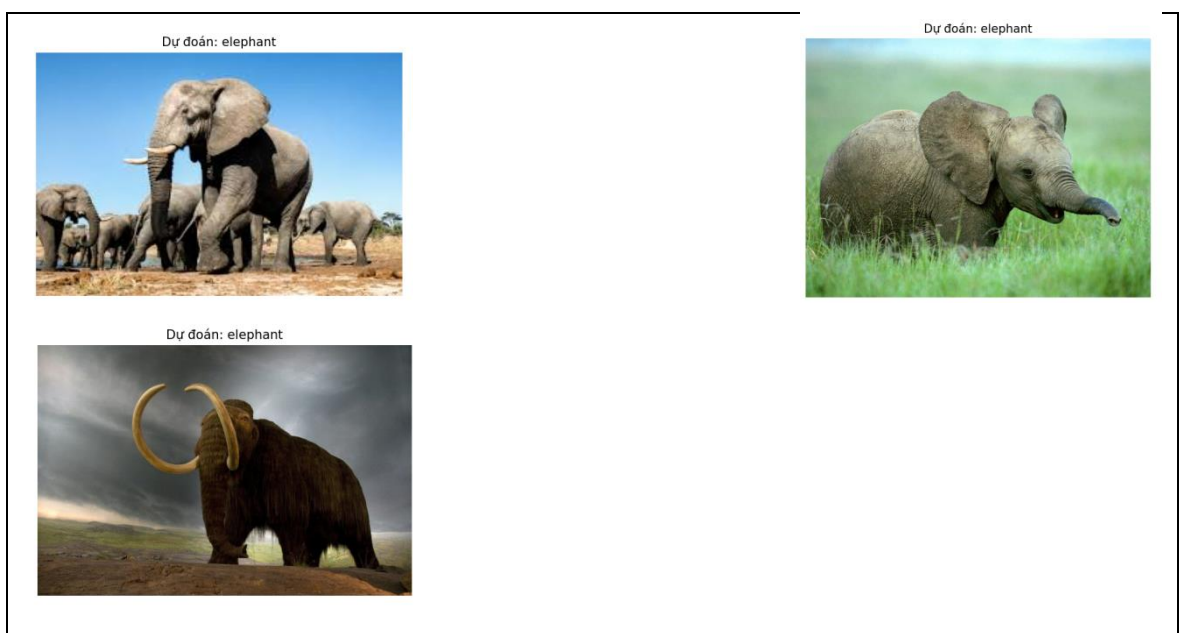
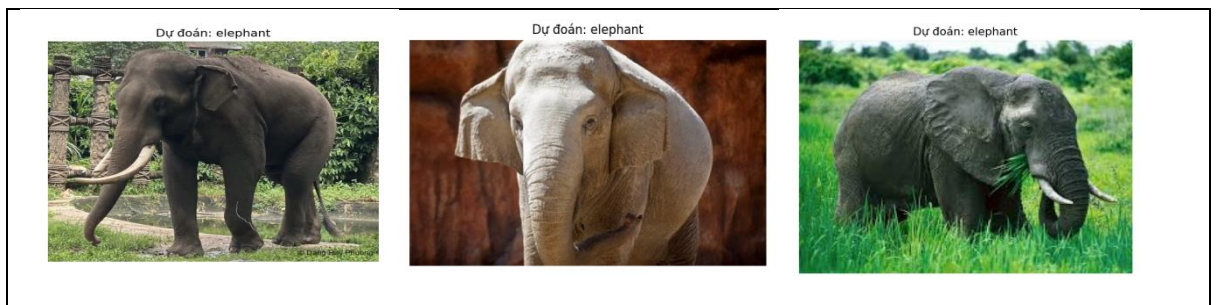






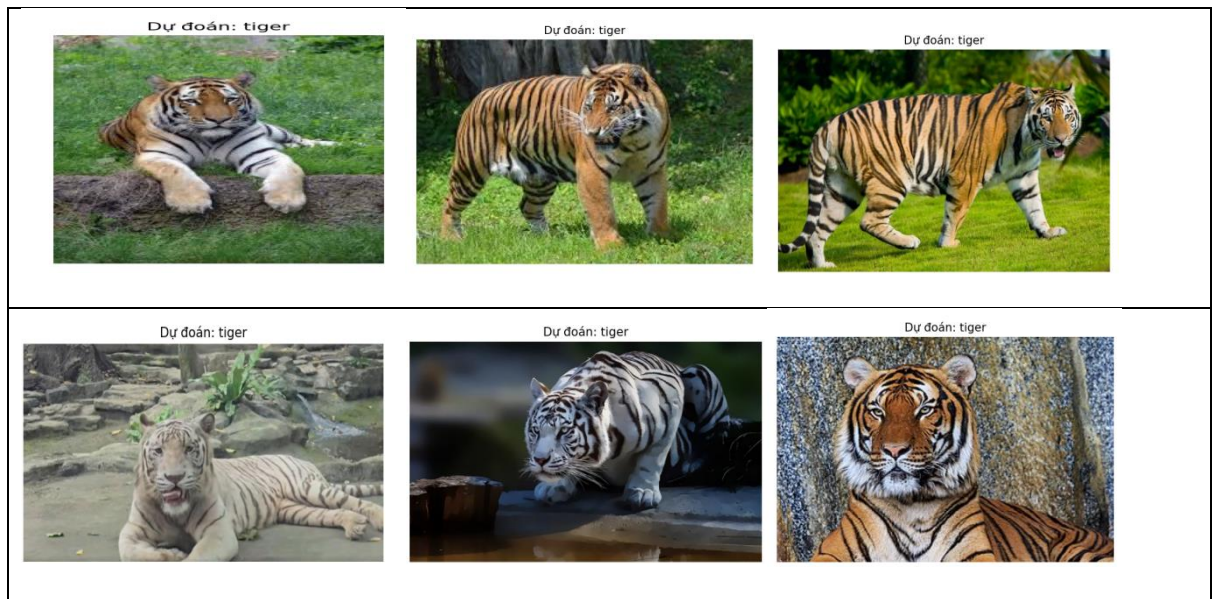
Hình 4. 6 Kết quả test ảnh dog trên tập train

Kết quả như ảnh elephant như sau:



Hình 4. 7 Kết quả test ảnh elephant trên tập train

Kết quả như ảnh tiger như sau:



Hình 4. 8 Kết quả test ảnh tiger trên tập train

Bảng thống kê kết quả test trên tập train:

Tên loài	Số ảnh dự đoán đúng	Số ảnh dự đoán sai	Tỷ lệ đúng
Cat	6	0	100%
Dog	6	0	100%
Elephant	6	0	100%
Tiger	6	0	100%

Bảng 4.2 Thống kê test 4 loài trên tập train

Trong bảng, qua các lần test, độ chính xác cao nhất là 100%, độ chính xác thấp nhất là 100%. Độ chính xác trung bình là 100%. So với kết quả test ở kịch bản trước, kịch bản này có độ chính xác rất cao. Điều này là do kiểm thử trên chính dữ liệu huấn luyện, nên kết quả tốt. Trong khi đó, kiểm thử trên dữ liệu mới hoàn toàn (trong kịch bản thứ nhất) thì kết quả chưa tốt do tập dữ liệu huấn luyện chưa đủ lớn để bao phủ hết các trường hợp. Nếu tăng dữ liệu huấn luyện, độ chính xác của mô hình sẽ được cải thiện. Vấn đề này sẽ được giải quyết trong “hướng phát triển tương lai”.

## 4.2 Đánh giá tổng thể mô hình

#### 4.2.1 Kết quả huấn luyện mô hình

Mô hình được huấn luyện trong 10 epoch với batch size = 32

Hàm mất mát categorical\_crossentropy và optimizer Adam.

Độ chính xác (accuracy) của mô hình tăng dần qua các epoch.

Giá trị loss giảm ổn định, cho thấy mô hình học hiệu quả và hội tụ tốt.

Điều này chứng tỏ kiến trúc CNN đã lựa chọn là phù hợp với tập dữ liệu và bài toán đặt ra.

#### 4.2.2 Kết quả trên tập kiểm tra (Test set)

Mô hình đạt độ chính xác tương đối tốt trên tập dữ liệu kiểm tra.

Không xuất hiện hiện tượng overfitting nghiêm trọng.

Sai số chủ yếu xuất hiện ở các ảnh có: Góc chụp phức tạp, nhiều vật thể trong cùng một ảnh, ánh sáng kém hoặc nền phức tạp. Điều này phản ánh hạn chế chung của các mô hình CNN khi dữ liệu huấn luyện còn hạn chế.

## CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1 Kết luận chung của đề tài

Sau quá trình nghiên cứu lý thuyết, tìm hiểu mô hình học sâu, chuẩn bị dữ liệu, xây dựng kiến trúc mạng và huấn luyện mô hình, đề tài “Xây dựng mô hình mạng nơ-ron tích chập (CNN) cho bài toán phân loại động vật” đã được hoàn thành đúng theo mục tiêu ban đầu đề ra. Mô hình CNN được xây dựng nhằm tự động nhận dạng và phân loại các loài động vật dựa trên ảnh đầu vào. Thông qua việc áp dụng các tầng tích chập, tầng gộp và tầng kết nối đầy đủ, mô hình có khả năng trích xuất đặc trưng ảnh và đưa ra kết quả phân loại tương ứng.

Đề tài không chỉ giúp sinh viên hiểu rõ hơn về nguyên lý hoạt động của CNN mà còn tạo cơ hội tiếp cận thực tế với quy trình xây dựng và triển khai một mô hình học sâu trong bài toán thị giác máy tính. Kết quả đạt được cho thấy mô hình hoạt động ổn định trong phạm vi nghiên cứu, đáp ứng được yêu cầu của đề án cơ sở ngành và có tính ứng dụng trong thực tế

### 5.2 Các kết quả nổi bật đạt được

Trong quá trình thực hiện đề án, đề tài đã đạt được một số kết quả nổi bật sau:

Xây dựng thành công mô hình mạng nơ-ron tích chập (CNN) để phân loại động vật dựa trên ảnh đầu vào.

Thực hiện đầy đủ các bước tiền xử lý dữ liệu ảnh gồm chia tập dữ liệu, thay đổi kích thước ảnh, chuẩn hóa giá trị pixel và gán nhãn tự động.

Thiết kế kiến trúc mô hình CNN gồm các tầng tích chập, max pooling, flatten và dense phù hợp với bài toán phân loại đa lớp.

Huấn luyện mô hình với các tham số phù hợp và đạt được kết quả phân loại tương đối tốt trên tập dữ liệu kiểm tra.

Triển khai chức năng dự đoán ảnh mới và hiển thị trực quan kết quả phân loại, giúp dễ dàng kiểm tra và đánh giá mô hình.

### 5.3 Ý nghĩa và giá trị đề tài

#### 5.3.1 Ý nghĩa học thuật

Đề tài giúp minh họa rõ ràng cách thức hoạt động của mô hình CNN trong bài toán phân loại ảnh, một trong những ứng dụng quan trọng của học sâu và trí tuệ nhân tạo. Thông qua việc xây dựng mô hình từ cơ bản, sinh viên có thể nắm vững các khái niệm về mạng nơ-ron tích chập, tiền xử lý dữ liệu ảnh và quy trình huấn luyện mô hình. Ngoài ra, đề tài còn góp phần củng cố kiến thức về lập trình Python, sử dụng thư viện TensorFlow/Keras và áp dụng lý thuyết học máy vào bài toán thực tế.

#### 5.3.2 Ý nghĩa thực tiễn

Mô hình CNN phân loại động vật có thể được ứng dụng trong nhiều lĩnh vực như giáo dục, nhận dạng hình ảnh, nghiên cứu sinh học hoặc các hệ thống thông minh.

Sản phẩm của đề tài có thể được sử dụng như một ví dụ minh họa trong giảng dạy hoặc là nền tảng để phát triển các ứng dụng nhận dạng hình ảnh nâng cao hơn trong tương lai.

### 5.4 Hạn chế của đề tài

Bên cạnh những kết quả đạt được, đề tài vẫn còn tồn tại một số hạn chế nhất định: Tập dữ liệu huấn luyện còn nhỏ, số lượng ảnh chưa đa dạng, ảnh hưởng đến độ chính xác tổng thể của mô hình. Chưa áp dụng các kỹ thuật tăng cường dữ liệu (Data Augmentation) để cải thiện khả năng tổng quát hóa.

Kiến trúc mô hình còn đơn giản, chưa sử dụng các mô hình CNN hiện đại và sâu hơn. Việc đánh giá mô hình chủ yếu dựa trên độ chính xác, chưa có các chỉ số nâng cao như confusion matrix hay precision, recall.

### 5.5 Hướng phát triển trong tương lai

Từ những hạn chế đã nêu, đề tài có thể được tiếp tục mở rộng và phát triển theo các hướng sau:

Mở rộng và đa dạng hóa tập dữ liệu huấn luyện để nâng cao độ chính xác của mô hình. Áp dụng các kỹ thuật Data Augmentation nhằm tăng tính đa dạng của dữ liệu ảnh. Sử dụng các mô hình CNN tiên tiến như VGG, ResNet, MobileNet thông qua Transfer Learning. Tối ưu các siêu tham số như số epoch, batch size và số tầng tích chập.

Triển khai mô hình thành ứng dụng web hoặc ứng dụng di động cho phép người dùng tải ảnh và nhận kết quả phân loại trực tiếp.

### 5.6 Kết luận cuối cùng

Nhìn chung, đề tài đã hoàn thành tốt vai trò của một đồ án cơ sở ngành cả về mặt học thuật lẫn sản phẩm thực tế. Mô hình CNN phân loại động vật không chỉ giúp sinh viên hiểu rõ hơn về học sâu và thị giác máy tính mà còn thể hiện khả năng vận dụng kiến thức chuyên ngành vào việc xây dựng một hệ thống trí tuệ nhân tạo có tính ứng dụng.

Kết quả của đồ án là nền tảng quan trọng để tiếp tục nghiên cứu, phát triển và hoàn thiện các hệ thống nhận dạng và phân loại hình ảnh trong tương lai.



## TÀI LIỆU THAM KHẢO

- [1] M. Crabbtree, “What is Machine Learning? Definition, Types, Tools & More,” 2023. [Online]. Available: <https://www.datacamp.com/blog/what-is-machine-learning>.
- [2] ibm, “What is deep learning?,” [Online]. Available: <https://www.ibm.com/think/topics/deep-learning>.
- [3] V. H. Tiệp, “Phân nhóm các thuật toán Machine Learning,” [Online]. Available: <https://machinelearningcoban.com/2016/12/27/categories/>.
- [4] A. Biswal, “Top 25 Deep Learning Applications Used Across Industries,” 2023. [Online]. Available: <https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-applications>.
- [5] Ibm, “What is neural network?,” [Online]. Available: <https://www.ibm.com/think/topics/neural-networks>.
- [6] Ibm, “What are convolutional neural network?,” [Online]. Available: <https://www.ibm.com/think/topics/convolutional-neural-networks>.
- [7] C. P. Van, “[Deep Learning] Tìm hiểu về mạng tích chập (CNN),” 2020. [Online]. Available: <https://viblo.asia/p/deep-learning-tim-hieu-ve-mang-tich-chap-cnn-maGK73bOKj2>.