

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG**

---



# **ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**Sinh viên** : Trần Hải Thành

**Giảng viên hướng dẫn:** TS. Ngô Trường Giang

**Hải Phòng -2023**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG**

-----

**ĐỀ TÀI : PHÂN LOẠI HÌNH ẢNH ĐỘNG VẬT  
SỬ DỤNG MẠNG NƠ-RON TÍCH CHẬP  
VÀ ỨNG DỤNG**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY  
NGÀNH : CÔNG NGHỆ THÔNG TIN**

**Sinh viên : Trần Hải Thành**

**Giảng viên hướng dẫn: TS. Ngô Trường Giang**

**Hải Phòng – 2023**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG**

-----

**NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP**

**Sinh viên** : Trần Hải Thành - **MSV** : 2012111008.

**Lớp** : CT2401C

**Ngành** : Công nghệ thông tin

**Tên đề tài** : Phân loại hình ảnh động vật sử dụng mạng nơ-ron tích chập  
và ứng dụng

## LỜI CẢM ƠN

Trong quá trình làm đồ án vừa qua vì được sự chỉ dẫn nhiệt tình của thầy TS. Ngô Trường Giang – Trường Đại học Thủy lợi, em đã hoàn thành đồ án của mình. Mặc dù em đã cố gắng với sự tận tâm của thầy, nhưng vì thời gian và khả năng nên đồ án của em vẫn còn không tránh được những điều thiếu sót.

- Em xin chân thành và bày tỏ lòng biết ơn sâu sắc đến thầy Ngô Trường Giang vì đã tận tình chỉ bảo, hướng dẫn và giành thời gian quý báu của mình cho em trong thời gian qua để em có thể hoàn thành đồ án của mình đúng thời hạn.
- Em xin cảm ơn tất cả thầy cô giáo trong khoa Công nghệ thông tin vì đã truyền đạt cho em rất các kiến thức nền tảng, chuyên ngành, chuyên môn và chuyên sâu cực kì vững chắc trong những năm qua để em có thể hoàn thành được đồ án này.

Em xin cảm ơn Trường Đại học Quản lý và Công nghệ Hải Phòng vì không ngừng hỗ trợ và tạo những điều kiện tốt nhất trong những năm vừa qua để em có thể học và thực hiện tốt đồ án.

Em xin cảm ơn gia đình, bạn bè đã hỗ trợ và cổ vũ cho em trong suốt quá trình học tập cũng như làm đồ án để em có thể hoàn thành khóa học và đồ án theo quy định.

Em xin chân thành cảm ơn!

## MỤC LỤC

<b>LỜI CẢM ƠN</b> .....	<b>1</b>
<b>DANH MỤC HÌNH MINH HỌA</b> .....	<b>7</b>
<b>MỞ ĐẦU</b> .....	<b>8</b>
<b>CHƯƠNG 1: TỔNG QUAN VỀ HỌC MÁY</b> .....	<b>9</b>
1.1 Giới thiệu về học máy (Machine Learning).....	9
1.1.1 Khái niệm .....	9
1.1.2 Nguyên tắc hoạt động .....	9
1.1.3 Ứng dụng của học máy trong thực tế.....	11
1.1.4 Ưu điểm và nhược điểm của học máy .....	13
1.2 Các phương pháp học máy.....	15
1.3 Giới thiệu về học sâu (Deep Learning).....	16
1.3.1 Khái niệm .....	16
1.3.2 Nguyên tắc hoạt động .....	17
1.3.3 Ứng dụng của học sâu trong thực tế .....	18
1.3.4 Ưu điểm và nhược điểm của học sâu .....	20
1.4 Bài toán phân lớp dữ liệu .....	22
1.4.1 Bài toán nhận dạng ảnh.....	23
1.4.2 Ứng dụng của bài toán nhận dạng hình ảnh.....	24
<b>CHƯƠNG 2: MẠNG NƠ-RON TÍCH CHẬP</b> .....	<b>26</b>
2.1 Giới thiệu về mạng nơ-ron.....	26
2.1.1 Khái niệm .....	26
2.1.2 Nguyên tắc hoạt động .....	26
2.1.3 Kiến trúc mạng nơ-ron .....	28
2.2 Giới thiệu về mạng nơ-ron tích chập (Convolutional Neural Network - CNN) .....	30
2.2.1 Khái niệm .....	30
2.2.2 Nguyên tắc hoạt động .....	30
2.2.3 Ứng dụng của mạng nơ-ron tích chập trong thực tế .....	32
2.2.4 Ưu điểm và nhược điểm của mạng nơ-ron tích chập.....	33

<b>CHƯƠNG 3: ỨNG DỤNG CỦA MẠNG NƠ-RON TÍCH CHẬP CHO PHÂN LỚP ẢNH .....</b>	<b>36</b>
3.1 Môi trường thử nghiệm.....	36
3.1.1 Giới thiệu về Google Colab .....	36
3.1.2 Cấu hình môi trường Google Colab.....	37
3.1.3 Các thư viện được sử dụng .....	38
3.2 Giới thiệu về ngôn ngữ lập trình Python .....	38
3.2.1 Khái niệm .....	38
3.2.2 Tại sao Python lại được sử dụng nhiều trong học máy?.....	40
3.3 Thu thập dữ liệu .....	41
3.4 Tiền xử lý dữ liệu.....	42
3.5 Xây dựng mô hình.....	43
3.5.1 Xây dựng tầng cho mô hình.....	43
3.5.2 Biên dịch và huấn luyện mô hình .....	45
3.6 Đánh giá mô hình.....	46
3.7 Một số kết quả thử nghiệm .....	46
3.7.1 Xây dựng hàm dự đoán 1 ảnh .....	46
3.7.2 Thử nghiệm với tập ảnh test.....	47
3.7.3 Thử nghiệm với tập ảnh huấn luyện .....	50
<b>KẾT LUẬN .....</b>	<b>54</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>56</b>

## DANH MỤC HÌNH MINH HỌA

Hình 1-1 Minh hoạ AI, học máy và học sâu .....	9
Hình 1-2 Mối quan hệ giữa học sâu với AI và học máy .....	17
Hình 1-3 Sự khác biệt giữa nguyên tắc hoạt động của học máy và học sâu... ..	18
Hình 1-4 Ví dụ mô tả quá trình phân loại thư rác .....	23
Hình 2-1 Quá trình hoạt động của mạng nơ-ron .....	28
Hình 2-2 Kiến Trúc mạng nơ-ron tích chập đơn giản.....	32
Hình 2-3 Ví dụ minh hoạ cấu trúc của một mạng nơ-ron tích chập .....	32
Hình 3-1 Các tiện ích của Google Colab .....	36
Hình 3-2 Logo ngôn ngữ lập trình Python.....	40
Hình 3-3 Một số ảnh mẫu.....	41
Hình 3-4 Thông tin chi tiết kiến trúc mô hình .....	44
Hình 3-5 Kết quả test_1 trên tập test.....	47
Hình 3-6 Kết quả test_2 trên tập test.....	47
Hình 3-7 Kết quả test_3 trên tập test.....	47
Hình 3-8 Kết quả test_4 trên tập test.....	47
Hình 3-9 Kết quả test_5 trên tập test.....	48
Hình 3-10 Kết quả test_6 trên tập test.....	48
Hình 3-11 Kết quả test_7 trên tập test.....	48
Hình 3-12 Kết quả test_8 trên tập test.....	48
Hình 3-13 Kết quả test_9 trên tập test.....	49
Hình 3-14 Kết quả test_10 trên tập test.....	49
Hình 3-15 Kết quả test_1 trên tập training.....	50
Hình 3-16 Kết quả test_2 trên tập training.....	50
Hình 3-17 Kết quả test_3 trên tập training.....	50
Hình 3-18 Kết quả test_4 trên tập training.....	51
Hình 3-19 Kết quả test_5 trên tập training.....	51
Hình 3-20 Kết quả test_6 trên tập training.....	51
Hình 3-21 Kết quả test_7 trên tập training.....	51
Hình 3-22 Kết quả test_8 trên tập training.....	51
Hình 3-23 Kết quả test_9 trên tập training.....	52
Hình 3-24 Kết quả test_10 trên tập training.....	52

## MỞ ĐẦU

Mô hình Deep Learning (Học Sâu) là một loại mô hình máy học sử dụng các mạng nơ-ron sâu để học các đặc trưng từ dữ liệu đầu vào. Mô hình Deep Learning có kiến trúc với nhiều tầng (lớp) nơ-ron, trong đó thông tin được truyền qua nhiều tầng để học các đặc trưng phân cấp từ dữ liệu. Mô hình học sâu được sử dụng trong nhiều lĩnh vực như thị giác máy tính (xử lý hình ảnh), xử lý ngôn ngữ tự nhiên, dịch máy, nhận diện tiếng nói, tự động lái xe, và nhiều ứng dụng khác. Deep Learning thường yêu cầu lượng dữ liệu lớn để huấn luyện hiệu quả và tránh overfitting (quá khớp).

Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến. Nó giúp xây dựng lên được những hệ thống thông minh với độ chính xác cao. Một trong những ứng dụng của CNN là nhận diện đối tượng trong hình ảnh. CNN được sử dụng để nhận diện các đối tượng trong hình ảnh, bao gồm các ứng dụng như nhận diện khuôn mặt, nhận diện đồ vật, nhận diện người dùng. Phân loại hình ảnh, CNN có thể phân loại các hình ảnh vào các lớp khác nhau.

Đề tài em chọn là sử dụng thuật toán CNN để phân loại hình ảnh động vật và ứng dụng, nội dung của đề án bao gồm:

Chương 1: Tổng quan về học máy.

Chương 2: Mạng nơ-ron nhân chập.

Chương 3: Ứng dụng CNN cho phân lớp ảnh.

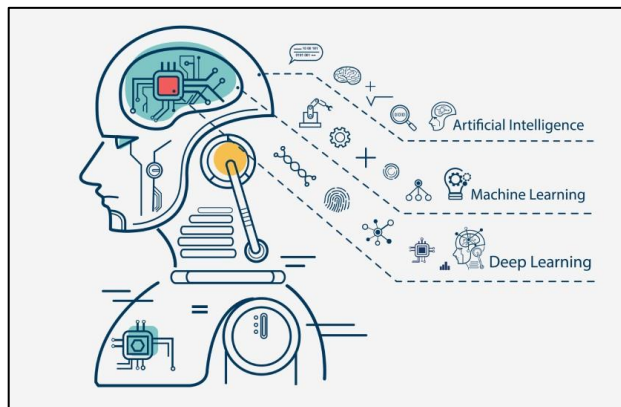


# CHƯƠNG 1: TỔNG QUAN VỀ HỌC MÁY

## 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 AI, học máy và học sâu**

### 1.1.2 Nguyên tắc hoạt động

Cách hoạt động của máy học bao gồm việc đi sâu vào quy trình từng bước để chuyển đổi dữ liệu thô thành những hiểu biết có giá trị. Quy trình này bao gồm một số bước chính sau:

#### **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...

- Dữ liệu cần được chuẩn bị theo định dạng phù hợp như là tập CSV, cơ sở dữ liệu và đảm bảo dữ liệu có liên quan đến vấn đề đang giải quyết.

### **Bước 2: Tiền xử lý dữ liệu**

- Tiền xử lý dữ liệu là một bước quan trọng nó xử lý dữ liệu (loại bỏ trùng lặp, sửa lỗi), xử lý dữ liệu bị thiếu (xóa hoặc điền vào) và chuẩn hóa 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.
- Các yếu tố cần xem xét bao gồm kích thước và loại dữ liệu, độ phức tạp của vấn đề và tài nguyên sẵn có.

### **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).

### **Bước 6: Điều chỉnh và tối ưu hóa siêu tham số**

- Điều chỉnh các siêu tham số của nó để cải thiện hiệu suất của nó. Sự tiến bộ này được gọi là điều chỉnh tham số hoặc tối ưu hóa siêu tham số. Các kỹ thuật điều chỉnh siêu tham số bao gồm tìm kiếm dạng lưới (thử các kết hợp tham số khác nhau) và xác thực chéo (thử chia dữ liệu của mình thành các tập hợp con và huấn luyện mô hình của mình trên từng tập hợp con để đảm bảo nó hoạt động tốt trên các dữ liệu khác nhau).

### **Bước 7: Dự đoán và triển khai**

- Sau khi mô hình được đào tạo và tối ưu hóa, nó sẵn sàng đưa ra dự đoán về dữ liệu mới. Quá trình này bao gồm việc cung cấp dữ liệu mới vào mô hình và sử dụng kết quả đầu ra của mô hình để đưa ra quyết định hoặc phân tích sâu hơn. Việc triển khai mô hình bao gồm việc tích hợp mô hình vào môi trường sản xuất nơi mô hình có thể xử lý dữ liệu trong thế giới thực và cung cấp thông tin chi tiết theo thời gian thực.

#### **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:

##### **1. Công nghệ thông tin và Internet:**

- Công nghệ tìm kiếm: Google, Bing và các công cụ tìm kiếm khác sử dụng ML để cải thiện kết quả tìm kiếm.

- Dịch thuật tự động: Các dịch vụ như Google Translate sử dụng ML để cải thiện hiệu suất dịch.
- Phân loại email: Các hộp thư đến tự động phân loại email vào các danh mục phù hợp bằng ML.

## **2. Tài chính:**

- Phân tích rủi ro và giao dịch: Ngân hàng và tổ chức tài chính sử dụng ML để phát hiện gian lận và phân tích rủi ro trong giao dịch tài chính.
- Dự báo thị trường: Công ty đầu tư sử dụng ML để dự đoán xu hướng thị trường và đánh giá rủi ro.

## **3. Y tế:**

- Chẩn đoán y học: ML được sử dụng để phát hiện các bệnh từ hình ảnh y khoa, như X-quang và MRI.
- Dự đoán bệnh: Các mô hình ML có thể được sử dụng để dự đoán nguy cơ mắc các bệnh lý dựa trên dữ liệu bệnh lý và di truyền.

## **4. Ô tô tự hành và IoT (Internet vạn vật):**

- Xe tự lái: Các hãng ô tô như Tesla và Waymo sử dụng ML để phát triển công nghệ xe tự lái.
- Cảm biến thông minh và quản lý năng lượng: ML được sử dụng trong các hệ thống quản lý năng lượng thông minh và cảm biến IoT.

## **5. Bán lẻ và Thương mại điện tử:**

- Tiếp thị cá nhân hóa: Các công ty bán lẻ sử dụng ML để tùy chỉnh trải nghiệm mua sắm cho từng khách hàng.
- Dự đoán nhu cầu và tồn kho: ML giúp dự đoán nhu cầu và quản lý tồn kho một cách hiệu quả.

## 6. Giáo dục:

- Học tập cá nhân hóa: Nền tảng giáo dục sử dụng ML để cá nhân hóa trải nghiệm học tập cho từng học viên.
- Đánh giá và phân loại học sinh: ML có thể giúp tự động đánh giá và phân loại bài làm của học sinh.

## 7. Quản lý chuỗi cung ứng:

- Dự đoán nhu cầu: ML được sử dụng để dự đoán nhu cầu sản phẩm và tối ưu hóa chuỗi cung ứng.
- Quản lý lưu trữ và vận chuyển: ML giúp cải thiện quy trình quản lý lưu trữ và vận chuyển hàng hóa.

## 8. Giải trí:

- Gợi ý nội dung: Các dịch vụ phát trực tuyến sử dụng ML để gợi ý nội dung dựa trên lịch sử xem phim hoặc âm nhạc của người dùng.
- Xử lý ngôn ngữ tự nhiên: ML được sử dụng để phát triển trí tuệ nhân tạo trong trò chơi video và các ứng dụng giải trí khác.

Đây chỉ là một số ví dụ, ML đang được áp dụng rộng rãi trong hầu hết các lĩnh vực của đời sống hàng ngày và còn nhiều ứng dụng khác đang được phát triển.

### 1.1.4 Ưu điểm và nhược điểm của học máy

Machine Learning (ML) có những ưu điểm và nhược điểm sau:

#### Ưu điểm:

1. Khả năng học từ dữ liệu: ML có khả năng tự học từ dữ liệu một cách tự động, giúp cải thiện hiệu suất và chính xác của các mô hình dự đoán và phân loại.

2. Khả năng xử lý dữ liệu lớn: ML có thể xử lý và phân tích lượng dữ liệu lớn một cách hiệu quả, giúp trích xuất thông tin quan trọng từ dữ liệu phức tạp.
3. Tính tự động hóa: ML có thể tự động hóa nhiều nhiệm vụ phức tạp mà trước đây cần sự can thiệp của con người, từ việc phân loại dữ liệu đến việc ra quyết định.
4. Tính linh hoạt và linh động: Các mô hình ML có thể được điều chỉnh và tinh chỉnh để phù hợp với yêu cầu cụ thể của mỗi ứng dụng, từ việc chọn lựa thuật toán đến việc điều chỉnh siêu tham số.
5. Áp dụng rộng rãi: ML có thể được áp dụng trong nhiều lĩnh vực khác nhau, từ công nghệ thông tin, y tế, tài chính đến giáo dục, sản xuất và nhiều lĩnh vực khác.

#### **Nhược điểm:**

1. Yêu cầu lượng dữ liệu lớn: Đa số các thuật toán ML đòi hỏi lượng dữ liệu lớn để huấn luyện mô hình một cách hiệu quả và đảm bảo tính chính xác.
2. Nguy cơ overfitting: Mô hình ML có thể mắc phải tình trạng overfitting, nghĩa là mô hình quá tinh chỉnh cho dữ liệu huấn luyện cụ thể mà không tổng quát hóa được cho dữ liệu mới.
3. Khả năng diễn giải kém: Một số mô hình ML phức tạp có thể khó hiểu và khó giải thích cách chúng ra quyết định, làm giảm tính minh bạch và tin cậy của mô hình.
4. Yêu cầu kiến thức chuyên sâu: Xây dựng và triển khai các mô hình ML đòi hỏi kiến thức vững về toán học, thống kê và lập trình, điều này có thể là một rào cản đối với người mới bắt đầu.

5. Nguy cơ quyền riêng tư và bảo mật: Một số ứng dụng ML có thể đối mặt với nguy cơ về quyền riêng tư và bảo mật khi sử dụng dữ liệu cá nhân hoặc nhạy cảm.

## 1.2 Các phương pháp học máy

Học máy là một phần của trí tuệ nhân tạo (AI) tập trung vào việc xây dựng các hệ thống có khả năng học và cải thiện từ dữ liệu một cách tự động, mà không cần phải được lập trình cụ thể cho mỗi nhiệm vụ. Thay vào đó, các thuật toán học máy sử dụng dữ liệu để phát hiện các mẫu, quy luật và hiểu biết từ kinh nghiệm, từ đó tạo ra các dự đoán hoặc hành động có ý nghĩa. [2]

Tóm lại, học máy cho phép máy tính học hỏi từ dữ liệu và trải nghiệm, từ đó tự động cải thiện hiệu suất của chúng trong các nhiệm vụ như nhận dạng ảnh, dự đoán, phân loại, gợi ý, và nhiều ứng dụng khác.

Cơ sở của Machine Learning nằm ở việc xử lý dữ liệu và tìm kiếm các mẫu và quy luật ẩn trong dữ liệu đó. Các phương pháp ML có thể được chia thành các loại chính bao gồm:

- **Học có giám sát (Supervised Learning):** Mô hình được huấn luyện trên một tập dữ liệu có nhãn, trong đó mỗi mẫu dữ liệu đã được gán một nhãn đích. Mục tiêu là để dự đoán hoặc phân loại dữ liệu mới dựa trên các quan hệ học từ dữ liệu huấn luyện.
- **Học không giám sát (Unsupervised Learning):** Trong trường hợp này, dữ liệu không có nhãn. Mô hình phải tự phát hiện ra cấu trúc, mẫu, hoặc nhóm trong dữ liệu mà không cần sự hướng dẫn từ người dùng.
- **Học bán giám sát (Semi-supervised Learning):** Khi chỉ một phần của dữ liệu được gán nhãn, và mục tiêu là để dự đoán hoặc phân loại các mẫu mới, sử dụng cả dữ liệu có nhãn và không nhãn.
- **Học tăng cường (Reinforcement Learning):** Mô hình học từ môi trường bằng cách thực hiện hành động và nhận phản hồi (thưởng hoặc

phạt) từ môi trường dựa trên các hành động đó. Mục tiêu của mô hình là tìm ra chuỗi hành động tối ưu để đạt được mục tiêu cụ thể.

Machine Learning có ứng dụng rộng rãi trong nhiều lĩnh vực, bao gồm nhận dạng ảnh và âm thanh, dịch văn bản, tự động lái xe, tư vấn y tế, dự báo thị trường tài chính, và nhiều lĩnh vực khác. Sự phát triển của Machine Learning đã đem lại nhiều tiềm năng và cơ hội lớn cho cả thế giới công nghệ và kinh doanh.

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

#### **1.3.1 Khái niệm**

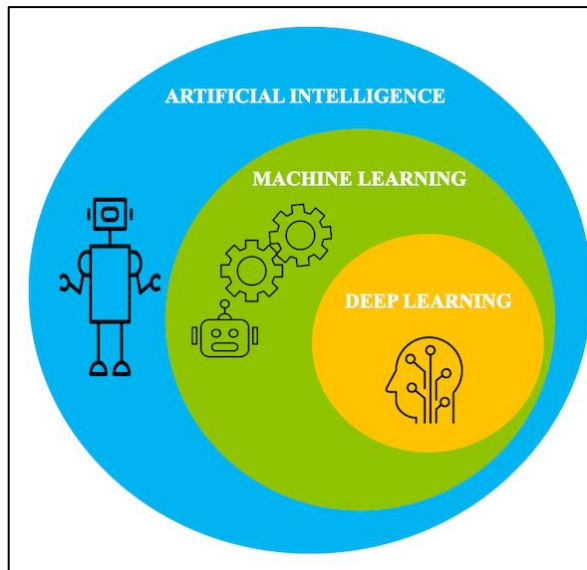
Học sâu (Deep Learning) là một phần của trí tuệ nhân tạo (AI) mà các mạng nơ-ron sâu được sử dụng để học và hiểu dữ liệu phức tạp. Trong Deep Learning, các mạng nơ-ron được cấu trúc với nhiều lớp (từ đó có tên gọi "sâu"), mỗi lớp thực hiện các phép tính phức tạp để tự động rút trích các đặc trưng từ dữ liệu. [3]

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.

Một trong những lợi ích lớn nhất của Deep Learning là khả năng học từ dữ liệu lớn một cách tự động và không cần sự can thiệp của con người để định rõ các đặc trưng hay quy tắc. Điều này giúp tạo ra các hệ thống thông minh có khả năng tự học và tự cải thiện theo thời gian.

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. [2]





**Hình 1-2** Mối quan hệ giữa học sâu với AI và học máy

### 1.3.2 Nguyên tắc hoạt động

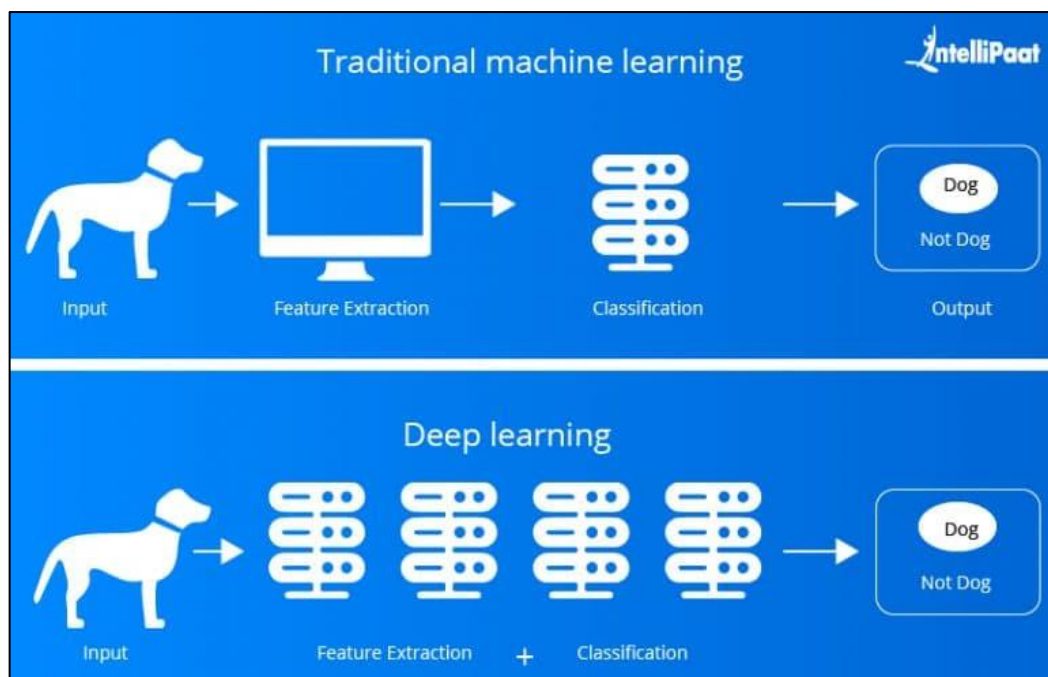
Mạng lưới thần kinh học sâu hoặc mạng lưới thần kinh nhân tạo cố gắng bắt chước bộ não con người thông qua sự kết hợp của dữ liệu đầu vào, trọng số và độ lệch. Các phần tử này phối hợp với nhau để nhận dạng, phân loại và mô tả chính xác các đối tượng trong dữ liệu.

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.

Một quy trình khác gọi là lan truyền ngược sử dụng các thuật toán, chẳng hạn như giảm độ dốc, để tính toán lỗi trong dự đoán, sau đó điều chỉnh trọng số và độ lệch của hàm bằng cách di chuyển ngược qua các lớp nhằm nỗ lực huấn luyện mô hình. Cùng với nhau, lan truyền thuận và lan truyền ngược cho phép mạng lưới thần kinh đưa ra dự đoán và sửa bất kỳ lỗi nào tương ứng. Theo thời gian, thuật toán dần trở nên chính xác hơn.

Phần 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.
- Mạng thần kinh tái phát (RNN) thường được sử dụng trong các ứng dụng nhận dạng giọng nói và ngôn ngữ tự nhiên vì nó tận dụng dữ liệu chuỗi thời gian hoặc tuần tự.



**Hình 1-3 Sự khác biệt giữa nguyên tắc hoạt động của học máy và học sâu**

### 1.3.3 Ứng dụng của học sâu trong thực tế

Deep Learning đã được áp dụng rộng rãi trong nhiều lĩnh vực khác nhau trong thực tế. Dưới đây là một số ứng dụng phổ biến của Deep Learning [4]:

## **1. Nhận dạng hình ảnh:**

- Nhận dạng đối tượng: Deep Learning được sử dụng để nhận dạng và phân loại các đối tượng trong hình ảnh, từ xe hơi đến động vật và đồ vật.
- Nhận dạng khuôn mặt: Deep Learning có thể xác định và phân biệt khuôn mặt của con người trong hình ảnh và video với độ chính xác cao.
- Xử lý hình ảnh y tế: Deep Learning giúp tự động phát hiện các bất thường trong hình ảnh y tế, như ung thư từ hình ảnh X-quang và MRI.

## **2. Xử lý ngôn ngữ tự nhiên (NLP):**

- Dịch thuật tự động: Deep Learning được sử dụng trong các dịch vụ dịch thuật tự động như Google Translate để cải thiện chất lượng dịch.
- Phân tích cảm xúc: Deep Learning có thể phân tích và hiểu cảm xúc từ văn bản, như xác định tâm trạng của một bình luận trên mạng xã hội.
- Xử lý ngôn ngữ tự nhiên trong ứng dụng trò chơi và chatbot: Deep Learning được sử dụng để phát triển chatbot thông minh và giao diện người dùng trong trò chơi.

## **3. Ô tô tự hành và xe thông minh:**

- Nhận diện biển báo giao thông: Deep Learning giúp xe tự hành nhận diện biển báo giao thông và thực hiện các hành động phản ứng phù hợp.
- Dự đoán hành vi người tham gia giao thông: Deep Learning có thể dự đoán hành vi của người tham gia giao thông, như dự đoán hướng đi của người đi bộ hoặc xe đạp.

- Xử lý dữ liệu từ cảm biến và camera: Deep Learning được sử dụng để xử lý dữ liệu từ cảm biến và camera trong xe tự hành và hệ thống xe thông minh.

#### **4. Y học và chăm sóc sức khỏe:**

- Chẩn đoán bệnh: Deep Learning giúp phân tích hình ảnh y học để hỗ trợ các bác sĩ trong việc chẩn đoán các bệnh lý.
- Dự đoán dị ứng và căn bệnh: Deep Learning có thể dự đoán nguy cơ mắc dị ứng và các căn bệnh dựa trên dữ liệu bệnh lý và di truyền.
- Phát hiện dấu hiệu sớm của bệnh: Deep Learning được sử dụng để phát hiện các dấu hiệu sớm của bệnh, giúp điều trị sớm và cải thiện tỷ lệ sống sót.

#### **5. Tổ chức và dự báo thị trường tài chính:**

- Dự đoán xu hướng thị trường: Deep Learning giúp dự đoán xu hướng thị trường tài chính và đưa ra các quyết định đầu tư thông minh.
- Phân tích tín hiệu giao dịch: Deep Learning được sử dụng để phân tích tín hiệu giao dịch và phát hiện các mô hình trong dữ liệu tài chính.

Đây chỉ là một số ví dụ, Deep Learning có rất nhiều ứng dụng trong nhiều lĩnh vực khác nhau và đang tiếp tục phát triển mạnh mẽ.

#### **1.3.4 Ưu điểm và nhược điểm của học sâu**

Deep Learning (Học sâu) là một phân nhánh của Machine Learning (ML) tập trung vào việc sử dụng các mạng nơ-ron sâu để học dữ liệu. Dưới đây là các ưu điểm và nhược điểm của Deep Learning:

### **Ưu điểm:**

1. Hiệu suất cao trong việc học dữ liệu phức tạp: Deep Learning có khả năng học được các đặc trưng phức tạp và trừu tượng từ dữ liệu đầu vào, giúp cải thiện hiệu suất của các mô hình dự đoán và phân loại.
2. Khả năng tự học đặc trưng: Deep Learning tự động học và trích xuất các đặc trưng từ dữ liệu, loại bỏ hoặc giảm thiểu sự phụ thuộc vào việc đặc trưng được xác định trước (feature engineering).
3. Tính linh hoạt và khả năng mở rộng: Deep Learning có thể áp dụng cho nhiều loại dữ liệu khác nhau như hình ảnh, văn bản, âm thanh và video. Nó cũng có thể mở rộng để xử lý các bài toán lớn và phức tạp.
4. Hiệu suất tốt trên dữ liệu lớn: Deep Learning có khả năng hiệu quả khi xử lý lượng dữ liệu lớn, đặc biệt là khi sử dụng các GPU hoặc TPU để tăng tốc quá trình huấn luyện.
5. Độ chính xác cao: Deep Learning thường có độ chính xác cao trong việc dự đoán và phân loại, đặc biệt là trong các bài toán nhận dạng hình ảnh và xử lý ngôn ngữ tự nhiên.

### **Nhược điểm:**

1. Yêu cầu lượng dữ liệu lớn: Deep Learning thường yêu cầu lượng dữ liệu lớn để huấn luyện mô hình một cách hiệu quả và tránh overfitting.
2. Yêu cầu tính toán cao: Quá trình huấn luyện mạng nơ-ron sâu đòi hỏi lượng tính toán lớn và thường cần sử dụng GPU hoặc TPU để tăng tốc.
3. Nguy cơ overfitting: Mạng nơ-ron sâu có nguy cơ cao về overfitting, đặc biệt là khi dữ liệu huấn luyện không đủ hoặc khi mô hình quá phức tạp.

4. Khó khăn trong việc diễn giải kết quả: Deep Learning thường tạo ra các mô hình phức tạp, khó diễn giải, làm giảm tính minh bạch và tin cậy của mô hình.
5. Yêu cầu kỹ năng và kiến thức chuyên sâu: Xây dựng và điều chỉnh các mạng nơ-ron sâu đòi hỏi kiến thức chuyên sâu về toán học, lý thuyết mạng nơ-ron và kỹ thuật lập trình.

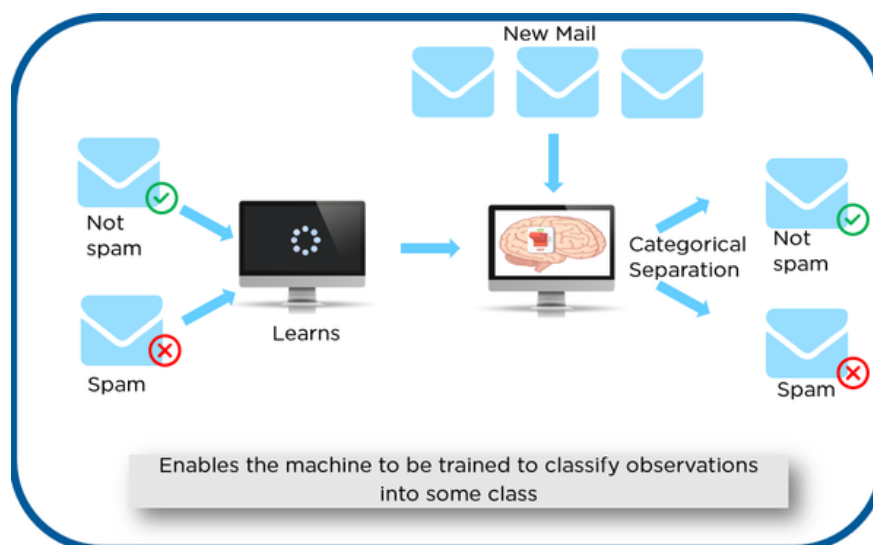
#### 1.4 Bài toán phân lớp dữ liệu

Bài toán phân lớp dữ liệu là một trong những vấn đề cơ bản trong Machine Learning, trong đó mục tiêu là phân loại các mẫu dữ liệu vào một trong các lớp hoặc nhóm đã được xác định trước. Mỗi mẫu dữ liệu thường được biểu diễn dưới dạng một vector đặc trưng, và mỗi lớp được gán một nhãn định danh.

Ví dụ, trong một bài toán phân loại email thành "spam" hoặc "không phải spam", dữ liệu đầu vào có thể là các email, và mỗi email được biểu diễn bằng một vector đặc trưng, có thể là số từ khóa xuất hiện trong email hoặc các đặc điểm khác. Mục tiêu của bài toán là xây dựng một mô hình máy học có khả năng phân loại các email mới vào một trong hai lớp "spam" hoặc "không phải spam" dựa trên các đặc trưng của chúng.

Trong bài toán phân lớp, quá trình huấn luyện mô hình thường bao gồm việc cung cấp cho mô hình một tập dữ liệu huấn luyện, trong đó mỗi mẫu dữ liệu đã được gán nhãn. Mô hình sẽ học từ các mẫu này để tìm ra các quy tắc hoặc mẫu trong dữ liệu, từ đó có thể phân loại các mẫu mới.

Các phương pháp phổ biến trong việc giải quyết bài toán phân lớp bao gồm các thuật toán như cây quyết định, máy vector hỗ trợ (SVM), học sâu (deep learning) và các thuật toán học máy khác. Đối với mỗi bài toán cụ thể, việc lựa chọn thuật toán thích hợp phụ thuộc vào tính chất của dữ liệu và yêu cầu của vấn đề.



**Hình 1-4 Ví dụ mô tả quá trình phân loại thư rác**

### 1.4.1 Bài toán nhận dạng ảnh

Bài toán nhận dạng ảnh, hay còn được gọi là bài toán nhận diện hình ảnh là một trong những ứng dụng phổ biến của trí tuệ nhân tạo và học máy. Mục tiêu chính của bài toán này là phân loại và xác định nội dung của hình ảnh dựa trên các đặc trưng và thông tin có sẵn trong hình ảnh.

Bài toán nhận dạng ảnh đặt ra yêu cầu với mô hình học sâu cần có khả năng hiểu và rút trích thông tin từ hình ảnh để phân loại chúng vào các nhãn hoặc lớp tương ứng. Để thực hiện điều này, mô hình học sâu cần được huấn luyện trên một tập dữ liệu lớn chứa các hình ảnh đã được gán nhãn.

Quá trình giải quyết bài toán nhận diện hình ảnh thường bao gồm các bước sau:

1. Chuẩn bị dữ liệu: Bước này bao gồm thu thập dữ liệu hình ảnh, resize ảnh gán nhãn cho chúng và chia thành tập huấn luyện và tập kiểm thử. Dữ liệu cần được chuẩn hóa và tiền xử lý trước khi đưa vào huấn luyện.
2. Xây dựng mô hình: Bước này liên quan đến việc xây dựng kiến trúc mô hình học sâu phù hợp với bài toán nhận dạng hình ảnh. Mô hình

thường được thiết kế với các lớp tích chập để trích xuất đặc trưng từ hình ảnh, sau đó là các lớp kết nối đầy đủ để thực hiện quá trình phân loại. Trong lĩnh vực nhận dạng ảnh, Convolutional Neural Network (CNN) thường là lựa chọn phổ biến.

3. Huấn luyện mô hình: Bước này là quá trình huấn luyện mô hình trên tập dữ liệu huấn luyện. Quá trình huấn luyện được thực hiện bằng cách cập nhật các trọng số của mô hình để giảm thiểu hàm mất mát (loss function) thông qua các phương pháp tối ưu hóa như gradient descent. Quá trình huấn luyện tiếp tục cho đến khi đạt được sự hội tụ hoặc một tiêu chí dừng khác.
4. Đánh giá mô hình: Bước này sử dụng tập dữ liệu kiểm thử để đánh giá hiệu suất của mô hình trên các dữ liệu mà nó chưa thấy. Đánh giá có thể được thực hiện bằng cách tính toán các độ đo.
5. Triển khai mô hình: Sử dụng mô hình đã huấn luyện để nhận dạng đối tượng trong ảnh mới hoặc trong môi trường thực tế.

Các ứng dụng của bài toán nhận dạng ảnh rất đa dạng, từ nhận dạng khuôn mặt, phân loại sản phẩm, đến nhận diện vật thể trong môi trường tự nhiên. Mô hình nhận dạng ảnh giúp tự động hóa và cải thiện nhiều quy trình trong nhiều lĩnh vực khác nhau

#### **1.4.2 Ứng dụng của bài toán nhận dạng hình ảnh**

Bài toán nhận dạng ảnh là một trong những 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: Nhận dạng khuôn mặt là một trong những ứng dụng phổ biến nhất của bài toán nhận dạng ảnh. Nó được sử dụng trong các hệ thống nhận dạng khuôn mặt để mở khóa điện thoại thông



minh, kiểm soát truy cập vào các cơ sở dữ liệu an ninh hoặc tạo ra các bộ lọc khuôn mặt trên các ứng dụng mạng xã hội.

2. Nhận dạng đối tượng: Bài toán nhận dạng đối tượng liên quan đến việc xác định và phân loại các đối tượng trong ảnh. Điều này có thể bao gồm việc nhận dạng xe hơi, động vật, đồ vật, hoặc bất kỳ đối tượng nào khác trong hình ảnh.
3. Phát hiện vật thể: Bài toán này tập trung vào việc xác định vị trí và phân loại vật thể trong ảnh. Điều này hữu ích trong các ứng dụng như xe tự lái, nơi cần xác định vị trí của các đối tượng như người đi bộ, ô tô và biển báo giao thông.
4. Nhận dạng biểu ngữ và chữ viết tay: Bài toán nhận dạng biểu ngữ và chữ viết tay giúp máy tính có thể đọc và hiểu nội dung của văn bản trong hình ảnh. Điều này có thể áp dụng trong việc tự động ghi chú từ hình ảnh hoặc dịch thuật tự động.
5. Nhận dạng vật thể và người trong video: Bài toán này tương tự như nhận dạng đối tượng, nhưng được áp dụng trên dữ liệu video. Nó được sử dụng trong các hệ thống giám sát an ninh để phát hiện hoạt động đáng ngờ hoặc nhận diện người trong video.
6. Nhận dạng y học: Bài toán này liên quan đến việc phân loại và nhận dạng các bệnh từ hình ảnh y học, như X-quang, MRI hoặc ảnh siêu âm, để hỗ trợ các chuyên gia y tế trong việc chẩn đoán và điều trị bệnh.

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: MẠNG NƠ-RON TÍCH CHẬP

### 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.

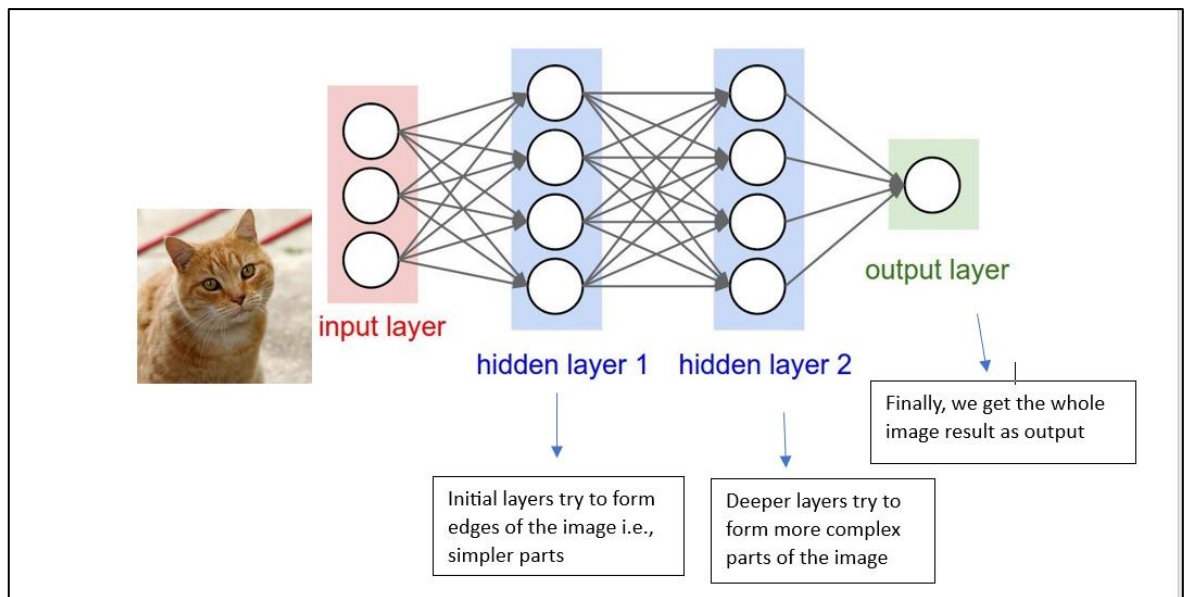
Mạng nơ-ron có khả năng học và tự điều chỉnh để thích nghi với dữ liệu, và chúng được sử dụng rộng rãi trong Machine Learning và Deep Learning để giải quyết các bài toán như nhận dạng hình ảnh, dịch thuật tự động, dự đoán dữ liệu, và nhiều ứng dụng khác.

#### 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.
7. 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 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.
- Thích hợp cho các bài toán phân loại đơn giản.

#### 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.

- Thích hợp cho các bài toán nhận dạng hình ảnh, phân loại văn bản, dự đoán chuỗi thời gian, và nhiều ứng dụng khác.

### **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 mạng nơ-ron tích chập (Convolutional Neural Network - CNN)

### 2.2.1 Khái niệm

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]

### 2.2.2 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. [6]

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]:

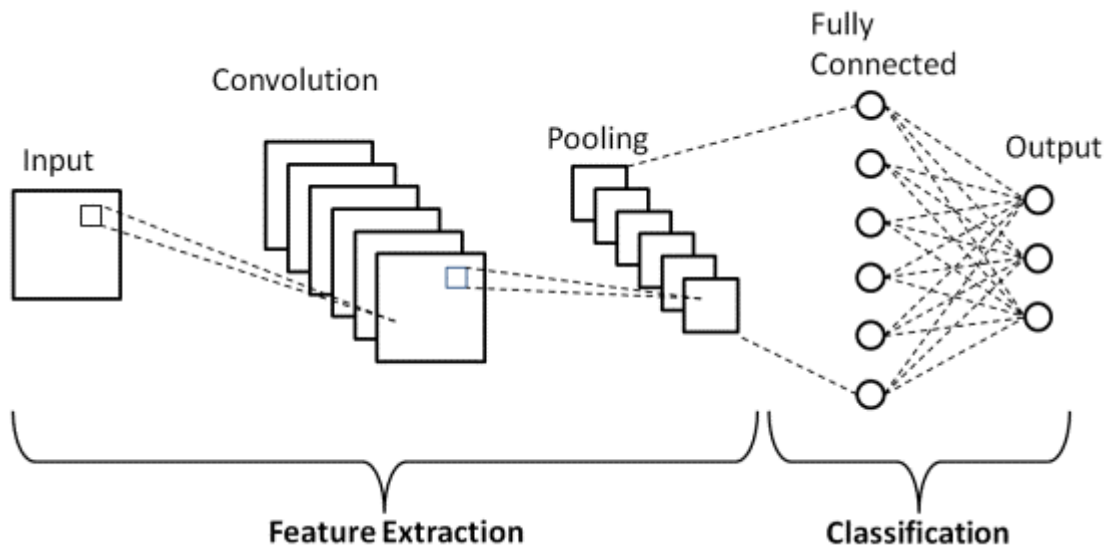
- **Đầu vào (Input):** Mạng CNN nhận vào dữ liệu đầu vào là các hình ảnh hoặc dữ liệu có cấu trúc dạng lưới khác như âm thanh, video.

- **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.

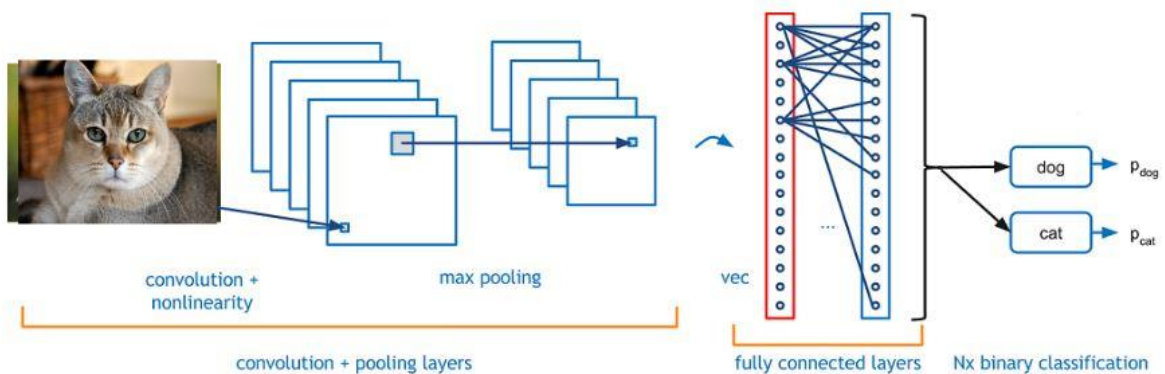
*\* Trong CNN, bản đồ đặc trưng (feature map) là đầu ra của lớp tích chập biểu thị các đặc trưng cụ thể của hình ảnh đầu.*

- **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.

Qua các bước này, mạng tích chập có khả năng học các đặc trưng cục bộ của dữ liệu đầu vào, giúp cho việc nhận dạng và phân loại trở nên hiệu quả hơn.



**Hình 2-2 Kiến Trúc mạng nơ-ron tích chập đơn giản**



**Hình 2-3 Ví dụ minh họa cấu trúc của một mạng nơ-ron tích chập**

### 2.2.3 Ứng dụng của 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 [8]:

- **Nhận diện và phân loại ảnh:** CNN được sử dụng để nhận diện và phân loại đối tượng trong ảnh, từ việc nhận diện khuôn mặt đến loại hoa, động vật, và vật thể khác.
- **Nhận diện và theo dõi video:** Trong lĩnh vực giám sát an ninh, CNN có thể giúp nhận diện hành vi đáng ngờ và theo dõi đối tượng trong video.



- **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

#### 2.2.4 Ưu điểm 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.

## CHƯƠNG 3: ỨNG DỤNG CỦA MẠNG NƠ-RON TÍCH CHẬP CHO PHÂN LỚP ẢNH

### 3.1 Môi trường thử nghiệm

#### 3.1.1 Giới thiệu về Google Colab

Google Colab (viết tắt của "Google Colaboratory") là một dịch vụ cung cấp môi trường làm việc miễn phí dựa trên đám mây của Google, cho phép viết và chạy mã Python trong môi trường Jupyter Notebook mà không cần cài đặt bất kỳ phần mềm nào trên máy tính cá nhân. Colab cung cấp một loạt các tính năng hữu ích bao gồm việc sử dụng các thư viện phổ biến như TensorFlow và PyTorch cho machine learning và deep learning, lưu trữ trên Google Drive, chia sẻ và làm việc cộng tác trực tuyến với người khác. Điều này làm cho nó trở thành một công cụ mạnh mẽ cho việc nghiên cứu và phát triển các dự án dựa trên Python mà không cần phải lo lắng về cấu hình phần cứng hoặc môi trường phát triển. [9]



**Hình 3-1** Các tiện ích của Google Colab

#### **Ưu điểm:**

- Miễn phí và dễ sử dụng: Colab là một dịch vụ miễn phí, giúp người dùng có thể thực hiện các dự án máy học mà không cần phải lo lắng về việc cài đặt và quản lý môi trường làm việc.

- Tích hợp với Google Drive: Colab tích hợp chặt chẽ với Google Drive, giúp lưu trữ và chia sẻ dễ dàng dự án.

- GPU và TPU hỗ trợ: Colab cung cấp truy cập vào GPU (Graphics Processing Unit) và TPU (Tensor Processing Unit) miễn phí, giúp gia tăng tốc độ huấn luyện mô hình machine learning.

- Thư viện và frameworks đã được cài đặt: Nhiều thư viện và frameworks phổ biến như TensorFlow, PyTorch, và OpenCV đã được cài sẵn, giúp giảm bớt bước đầu cài đặt môi trường làm việc.

- Hỗ trợ jupyter notebook: Colab sử dụng định dạng Jupyter Notebook, giúp người dùng tạo, chia sẻ và lưu trữ dễ dàng các tệp notebook của mình.

### **Nhược điểm:**

- Giới hạn tài nguyên: Người dùng miễn phí có giới hạn thời gian sử dụng GPU và TPU.

- Không đảm bảo ổn định và duy trì dài hạn: Do là dịch vụ miễn phí, không có sự đảm bảo về sự ổn định và tính khả dụng so với dịch vụ trả phí

### **3.1.2 Cấu hình môi trường Google Colab**

Huấn luyện (hay còn gọi là train) một mô hình Deep Learning, cần xử lý lượng phép tính lớn hơn nhiều so với các mô hình Machine Learning khác. Để cải thiện tốc độ tính toán, GPU (Graphics Processing Unit) được dùng thay cho CPU (Central Processing Unit), vì GPU cho phép xử lý nhiều phép tính song song với rất nhiều core sẽ nhanh hơn nhiều so với CPU. Tuy nhiên giá của GPU thì khá đắt đỏ để mua hoặc thuê server có GPU. Thế nên Google đã cung cấp Google Colab miễn phí có GPU để chạy code python (deep learning) cho mục đích nghiên cứu.

Sau khi đăng nhập tài khoản Google, tạo notebook thông qua trang web của Google Colab, thay đổi từ CPU sang GPU thông qua Runtime → Change runtime type → T4 GPU → Save. [9]

### 3.1.3 Các thư viện được sử dụng

OS: Đây là thư viện để thao tác tệp và thư mục.

Numpy và Pandas: Đây là thư viện quan trọng trong xử lý dữ liệu và tính toán số học. Numpy hỗ trợ các phép toán số học trên mảng, pandas được sử dụng để làm việc với dữ liệu dạng bảng (DataFrame).

Matplotlib: Đây là thư viện trực quan hóa dữ liệu. Matplotlib thường được sử dụng cho các biểu đồ cơ bản.

CV2: Đây là thư viện xử lý hình ảnh của OpenCV.

PIL: Đây là thư viện xử lý hình ảnh của Python.

TensorFlow / Keras: Đây là các thư viện mã nguồn mở được phát triển bởi Google, được sử dụng để xây dựng các mô hình học máy và mạng nơ-ron.

Sequential: Đây là một trong các phương thức từ keras để xây dựng mô hình. Sequential được sử dụng để xây dựng mô hình theo kiểu tuần tự.

ImageDataGenerator: Đây là một phương pháp trong keras giúp tạo ra các biến thể của ảnh từ dữ liệu huấn luyện, giúp tăng cường hình ảnh (Image augmentation) và tránh overfitting.

Conv2D, MaxPooling2D, BatchNormalization, Dense, Dropout, Flatten : Đây là các lớp liên quan đến việc xây dựng mạng CNN.

ReduceLROnPlateau: Đây là “Callback”. Callback trong Keras là các hàm có thể thực hiện các hành động cụ thể tại các giai đoạn khác nhau trong quá trình huấn luyện. Nếu hiệu suất của model không cải thiện thì có thể giảm tỷ lệ học đi để tránh overfitting.

## 3.2 Giới thiệu về ngôn ngữ lập trình Python

### 3.2.1 Khái niệm

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. Python có

thể được sử dụng trong nhiều lĩnh vực, từ phát triển web đến khoa học dữ liệu và trí tuệ nhân tạo.

Những lợi ích của Python:

- Dễ dàng đọc và hiểu.
- Giúp cải thiện năng suất làm việc.
- Thư viện tiêu chuẩn phong phú.
- Có thể tích hợp và tương tác một cách dễ dàng với các ngôn ngữ lập trình khác.
- Có một cộng đồng lớn và nhiệt tình trên toàn thế giới.
- Trên Internet có rất nhiều tài nguyên hữu ích về Python.
- Có thể sử dụng trên nhiều hệ điều hành khác nhau

Đặc điểm của Python:

- Python là một ngôn ngữ thông dịch.
- Python là một ngôn ngữ dễ sử dụng.
- Python là một ngôn ngữ linh hoạt.
- Python là một ngôn ngữ cấp cao.
- Python là một ngôn ngữ lập trình hướng đối tượng.

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



**Hình 3-2 Logo ngôn ngữ lập trình Python**

### **3.2.2 Tại sao Python lại được sử dụng nhiều trong học máy?**

Mọi người thường sử dụng Python trong lĩnh vực Machine Learning vì có một số lý do sau:

1. Dễ học và dễ hiểu: Python được thiết kế với một cú pháp rõ ràng và dễ đọc, giúp người mới bắt đầu nắm bắt cú pháp nhanh chóng và dễ dàng.
2. Cộng đồng lớn và hỗ trợ tốt: Python có một cộng đồng lập trình viên rất lớn và nhiều người dùng trên toàn thế giới. Do đó, nếu gặp phải vấn đề hoặc cần sự giúp đỡ, có thể dễ dàng tìm kiếm thông tin trên Internet hoặc nhờ sự hỗ trợ từ cộng đồng.
3. Các thư viện và framework phong phú: Python có nhiều thư viện và framework mạnh mẽ hỗ trợ Machine Learning như NumPy, Pandas, Scikit-learn, TensorFlow và PyTorch. Những thư viện này cung cấp các công cụ và tính năng cần thiết để phát triển và triển khai các mô hình Machine Learning một cách dễ dàng và hiệu quả.



4. Tích hợp tốt với các công nghệ khác: Python có khả năng tích hợp tốt với các công nghệ khác, bao gồm các thư viện và framework được sử dụng rộng rãi trong các lĩnh vực khác như web development, data analysis và automation. Điều này làm cho việc xây dựng và triển khai các ứng dụng Machine Learning trở nên dễ dàng hơn.
5. Sự linh hoạt và mở rộng: Python là một ngôn ngữ linh hoạt và có khả năng mở rộng cao. Python có thể sử dụng cho nhiều mục đích khác nhau trong Machine Learning, từ việc xử lý và làm sạch dữ liệu đến việc xây dựng và đào tạo mô hình, và thậm chí triển khai các ứng dụng Machine Learning vào sản phẩm thực tế.

### 3.3 Thu thập dữ liệu

Bộ dữ liệu “Dog-vs-Cat” được lấy từ trên Kaggle bao gồm 9896 bức ảnh chia thành 2 tập là tập huấn luyện và tập kiểm thử, trong đó 5011 bức ảnh là ảnh con mèo 4885 bức ảnh là ảnh con chó. Ảnh con mèo được gán nhãn thông qua đặt tên file là “cat.x” đặt trong thư mục cats, và ảnh con chó được gán nhãn thông qua đặt tên file là “dog.x” đặt trong thư mục dogs. Hình sau minh họa một số ảnh được gán nhãn. [10]



**Hình 3-3 Một số ảnh mẫu**

Tập dữ liệu huấn luyện được tổ chức riêng trong 2 thư mục: `training_set` (tập huấn luyện) và thư mục `test_set` (tập kiểm thử). 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`.

### 3.4 Tiền xử lý dữ liệu

Dữ liệu huấn luyện và kiểm thử được tiền xử lý trước khi đưa vào mô hình để đảm bảo dữ liệu đầu vào phù hợp với các yêu cầu khi sử dụng thư viện. Việc tiền xử lý bao gồm các bước sau:

- Chuẩn hóa dữ liệu về khoảng 0..1. Ảnh sử dụng là ảnh RGB, mỗi điểm ảnh trong ảnh sẽ có giá trị từ khoảng 0 đến 255. Việc chuẩn hóa được thực hiện với hàm `ImageDataGenerator` trong thư viện TensorFlow như sau:

```
# Các tham số
image_size = 128 # Cỡ của ảnh
image_channel = 3 # Số kênh màu (RGB)
bat_size = 32 # Số file/ảnh được xử lý tại mỗi lần
# Tạo dữ liệu
# Tiền xử lý ảnh ==> chuẩn hóa giá trị mức xám về dạng 0..1
train_datagen = ImageDataGenerator(rescale=1./255)

test_datagen = ImageDataGenerator(rescale=1./255)
```

- Tạo dữ liệu huấn luyện và kiểm thử từ thư mục ảnh đã có. Bước này nhằm mục đích tạo ra dữ liệu đầu vào theo cấu trúc gồm 2 phần: dữ liệu ảnh và nhãn với các thông số sau

```
train_gen=
train_datagen.flow_from_directory('/content/drive/MyDrive/Dog-vs-
Cat/data/training_set/',
class_mode='categorical',
target_size = (image_size,image_size),
batch_size = bat_size,
)

val_gen=
test_datagen.flow_from_directory('/content/drive/MyDrive/Dog-vs-
Cat/data/test_set/',
class_mode='categorical',
```

```
batch_size = bat_size,
target_size = (image_size, image_size),
shuffle = False
)
```

## 3.5 Xây dựng mô hình

### 3.5.1 Xây dựng tầng cho mô hình

Tạo một mô hình tuần tự rỗng, gồm các tầng và thành phần trong các tầng như sau:

- Tầng input bao gồm lớp tích chập Conv2D với 32 bộ lọc và hàm kích hoạt relu.
- Mô hình có thêm một tầng ẩn chứa lớp tích chập với các bộ lọc tăng dần và hàm kích hoạt relu.
- Mỗi tầng chứa lớp tích chập có thêm lớp Batch Normalization (giảm thiểu overfitting), lớp Max pooling và lớp Dropout (ngăn chặn overfitting).
- Tầng kết nối đầy đủ bao gồm lớp Flatten, lớp Dense và lớp Dropout.
- Tầng output chứa lớp Dense và hàm kích hoạt softmax (dự đoán lớp của ảnh đầu vào).

Sử dụng thư viện Keras, thông tin chi tiết về kiến trúc mô hình sẽ được thể hiện ra như sau:

```

model.summary()

Model: "sequential_5"

```

Layer (type)	Output Shape	Param #
conv2d_10 (Conv2D)	(None, 126, 126, 32)	896
batch_normalization_15 (Batch Normalization)	(None, 126, 126, 32)	128
max_pooling2d_10 (MaxPooling2D)	(None, 63, 63, 32)	0
dropout_15 (Dropout)	(None, 63, 63, 32)	0
conv2d_11 (Conv2D)	(None, 61, 61, 64)	18496
batch_normalization_16 (Batch Normalization)	(None, 61, 61, 64)	256
max_pooling2d_11 (MaxPooling2D)	(None, 30, 30, 64)	0
dropout_16 (Dropout)	(None, 30, 30, 64)	0
flatten_5 (Flatten)	(None, 57600)	0
dense_10 (Dense)	(None, 512)	29491712
batch_normalization_17 (Batch Normalization)	(None, 512)	2048
dropout_17 (Dropout)	(None, 512)	0
dense_11 (Dense)	(None, 2)	1026

```

Total params: 29514562 (112.59 MB)
Trainable params: 29513346 (112.58 MB)
Non-trainable params: 1216 (4.75 KB)

```

### Hình 3-4 Thông tin chi tiết kiến trúc mô hình

Đoạn mã sau minh họa việc tạo mô hình

```

model = Sequential()

# Tầng Input
model.add(Conv2D(32, (3, 3), activation='relu', input_shape =
(image_size, image_size, image_channel)))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))

# Tầng ẩn 1
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))

```

```

# Tầng kết nối đầy đủ
model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.2))

# Tầng Output
model.add(Dense(2, activation='softmax'))

model.summary()

```

### 3.5.2 Biên dịch và huấn luyện mô hình

Mô hình trước khi có thể huấn luyện cần phải được biên dịch. Các thông số cần thiết cho quá trình này bao gồm bộ tối ưu, hàm mất mát và độ đo:

- **Optimizer:** Bộ tối ưu (Optimizer) được sử dụng là Adam. Adam (viết tắt của Adaptive Moment Estimation) là một thuật toán mạnh mẽ kết hợp tốc độ học thích ứng, hiệu chỉnh sai lệch và động lượng để tối ưu hóa hiệu quả quá trình đào tạo mạng nơ-ron.
- **Loss:** Muốn làm cho mô hình tốt hơn, phải giảm thiểu tổn thất hoặc tối đa hóa độ chính xác. Để tính toán được chúng trong vấn đề này, dùng phương thức “binary\_crossentropy” (vì lớp đầu ra của chỉ có 2 lựa chọn)
- **Metrics:** Độ đo được sử dụng để đánh giá mức độ hiệu quả của mô hình trong việc dự đoán nhãn lớp một cách chính xác.

Đoạn mã sau thực hiện biên dịch mô hình

```

#Hàm callback
learning_rate_reduction = ReduceLRonPlateau(monitor = 'val_accuracy',
                                             patience=2,
                                             factor=0.5,
                                             min_lr = 0.00001,
                                             verbose = 1)

#Biên dịch mô hình
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

#Huấn luyện mô hình
cat_dog = model.fit(train_gen,
                    validation_data = val_gen,
                    callbacks=[learning_rate_reduction],
                    epochs = 10,
                    # steps_per_epoch = len(train_generator),

```

```

        # validation_steps = len(val_generator),
    )
# Lưu mô hình
model.save("/content/drive/MyDrive/Dog-vs-Cat/model.h5")

```

### 3.6 Đánh giá mô hình

Mô hình được đánh giá sử dụng phương thức evaluate của mô hình được xây dựng trong thư viện tensorflow. Kết quả cho thấy mô hình đạt độ chính xác 75,86% trên tập dữ liệu thử nghiệm. Tỷ lệ này chưa thật sự tốt trên bộ dữ liệu đánh giá do sự hạn chế về mặt cấu hình và số lượng dữ liệu huấn luyện.

```

# Evaluate đánh giá độ chính xác của mô hình đã xây dựng và huấn luyện trên dữ liệu test: test_generator
loss,acc = model.evaluate(test_generator, batch_size = bat_size, verbose = 0)

print('The accuracy of the model for testing data is:',acc*100)
print('The Loss of the model for testing data is:',loss)

The accuracy of the model for testing data is: 75.86206793785095
The Loss of the model for testing data is: 0.8027000427246094

```

### 3.7 Một số kết quả thử nghiệm

#### 3.7.1 Xây dựng hàm dự đoán 1 ảnh

Input: đường dẫn đến file ảnh cần dự đoán

Output: Gán nhãn cho ảnh dự đoán (Cat hoặc dog)

```

#Xây dựng hàm để dự đoán cho 1 ảnh
def pred_single(path):
    img = imread(path)
    resized_image = cv2.resize(img, (image_size, image_size))
    resized_image = np.expand_dims(resized_image, axis=0)
    rescaled_image = resized_image / 255.0

    # Tạo hàm dự đoán
    predictions = model.predict(rescaled_image)

    # Trích xuất lớp có xác suất cao nhất
    predicted_class = np.argmax(predictions)

    if predicted_class == 0:
        pred = 'cat'
    else:
        pred='dog'

    return [img, pred]

```



### 3.7.2 Thử nghiệm với tập 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 `x_test`. Thử nghiệm sẽ tiến hành 10 test, mỗi test lấy ngẫu nhiên 5 ảnh để dự đoán, kết quả như sau:



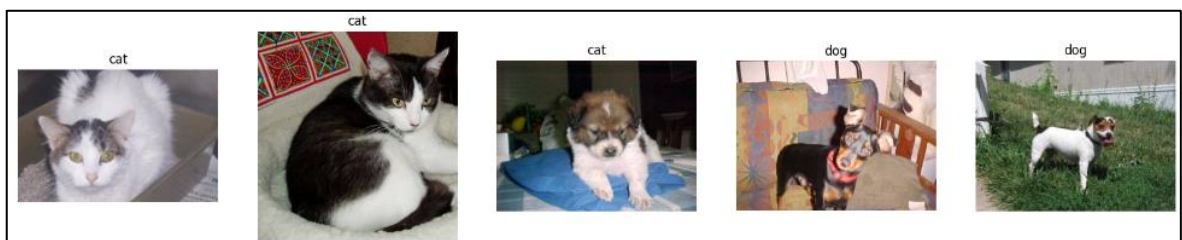
Hình 3-5 Kết quả test\_1 trên tập test



Hình 3-6 Kết quả test\_2 trên tập test



Hình 3-7 Kết quả test\_3 trên tập test



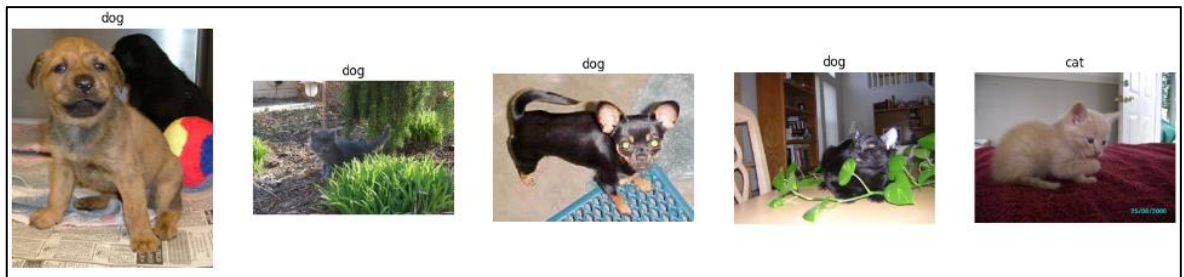
Hình 3-8 Kết quả test\_4 trên tập test



**Hình 3-9 Kết quả test\_5 trên tập test**



**Hình 3-10 Kết quả test\_6 trên tập test**



**Hình 3-11 Kết quả test\_7 trên tập test**



**Hình 3-12 Kết quả test\_8 trên tập test**





**Hình 3-13 Kết quả test\_9 trên tập test**



**Hình 3-14 Kết quả test\_10 trên tập test**

Bảng thống kê kết quả test:

Test	Số ảnh dự đoán đúng	Số ảnh dự đoán sai	Tỷ lệ đúng
Test_1	4	1	80%
Test_2	3	2	60%
Test_3	3	2	60%
Test_4	4	1	80%
Test_5	3	2	60%
Test_6	3	2	60%
Test_7	3	2	60%
Test_8	3	2	60%
Test_9	3	2	60%
Test_10	3	2	60%

Trong bảng, qua các lần test, test\_x có độ chính xác cao nhất là 80%, test\_x có độ chính xác thấp nhất là 60%. Độ chính xác trung bình của 10 test là 64%.

### 3.7.3 Thử nghiệm với tập ảnh huấn luyện

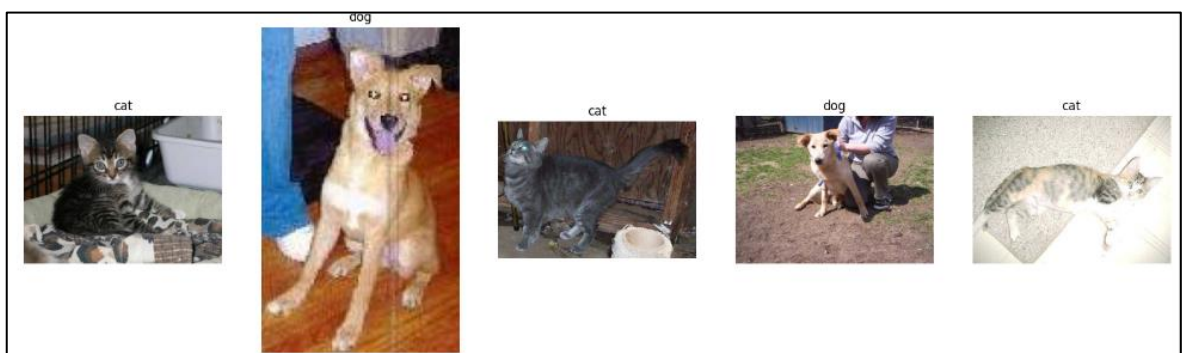
Thử nghiệm này được thực hiện trên các ảnh nằm trong tập huấn luyện. Kịch bản test cũng được thực hiện như với thử nghiệm trước bao gồm 10 test, mỗi test lấy ngẫu nhiên 5 ảnh để dự đoán. Kết quả như sau:



**Hình 3-15 Kết quả test\_1 trên tập training**



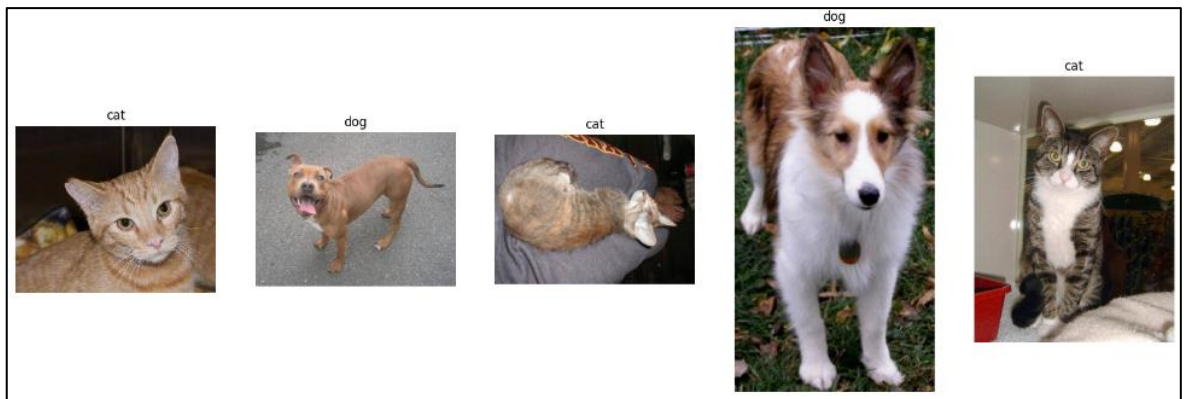
**Hình 3-16 Kết quả test\_2 trên tập training**



**Hình 3-17 Kết quả test\_3 trên tập training**



**Hình 3-18 Kết quả test\_4 trên tập training**



**Hình 3-19 Kết quả test\_5 trên tập training**



**Hình 3-20 Kết quả test\_6 trên tập training**

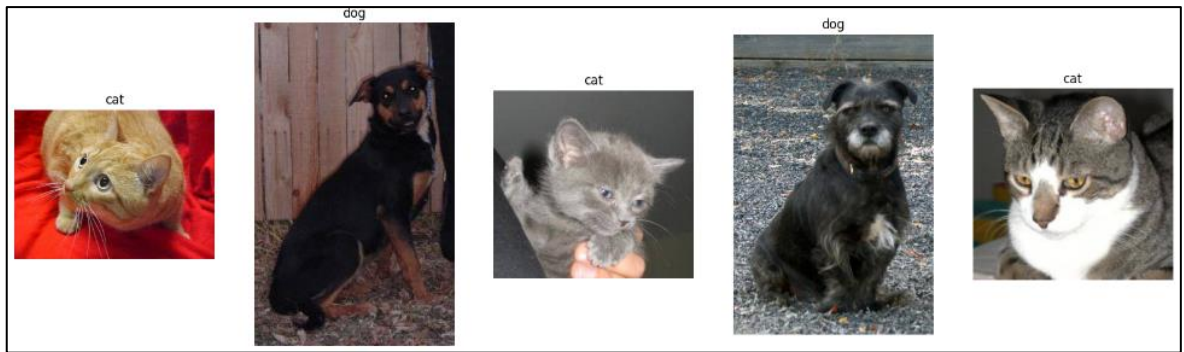


**Hình 3-21 Kết quả test\_7 trên tập training**

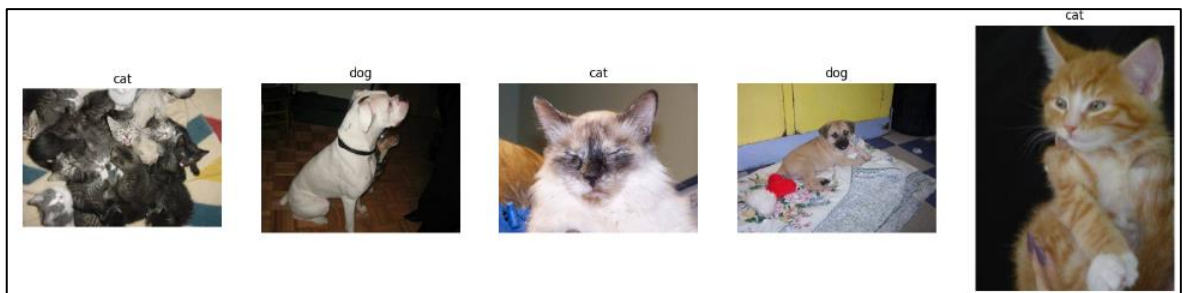


**Hình 3-22 Kết quả test\_8 trên tập training**





**Hình 3-23 Kết quả test\_9 trên tập training**



**Hình 3-24 Kết quả test\_10 trên tập training**

Bảng thống kê kết quả test:

Test	Số ảnh dự đoán đúng	Số ảnh dự đoán sai	Tỷ lệ đúng
Test_1	5	0	100%
Test_2	5	0	100%
Test_3	5	0	100%
Test_4	5	0	100%
Test_5	5	0	100%
Test_6	5	0	100%
Test_7	5	0	100%
Test_8	5	0	100%
Test_9	5	0	100%
Test_10	5	0	100%

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 của 10 test 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”.

## KẾT LUẬN

Thông qua đồ án tốt nghiệp với đề tài “Sử dụng thuật toán CNN để phân loại hình ảnh động vật” em đã có được một số kết quả và có một số hạn chế như sau:

Kết quả đạt được:

- Tìm hiểu tổng quan về Machine learning và Deep learning.
- Tìm hiểu về ngôn ngữ lập trình Python, chương trình chạy phụ trợ TensorFlow.
- Tìm hiểu về mạng CNN.
- Xây dựng mô hình CNN ứng dụng trong nhận diện động vật.
- Đánh giá độ chính xác của mô hình.
- Thử nghiệm mô hình trong nhận diện động vật.

Một số hạn chế:

- Trong quá trình huấn luyện mô hình thời gian chạy khá lâu nên có xảy ra một số sự cố như đứng màn hình và phải chạy huấn luyện lại mô hình rất nhiều lần
- Số lượng ảnh huấn luyện còn ít, bên cạnh đó số lần lặp trong quá trình huấn luyện bị giới hạn do năng lực tính toán của máy tính hạn chế không đáp ứng được. Do vậy độ chính xác còn thấp.
- Mới chỉ dừng lại thử nghiệm xây dựng mô hình CNN đơn giản, chưa so thử nghiệm và sánh với các mô hình hiện đại để tìm ra mô hình phù hợp nhất cho tập dữ liệu này

Hướng phát triển

- Thử nghiệm và đánh giá với nhiều kiến trúc CNN khác nhau để tìm mô hình phù hợp nhất

- Tăng số lượng ảnh huấn luyện và sử dụng máy tính có năng lực tính toán cao hơn để tăng độ chính xác của mô hình

Mặc dù cố gắng trong quá trình thực hiện DATN để hoàn thiện đồ án một cách tốt nhất nhưng do năng lực, kiến thức và trình độ còn nhiều hạn chế nên em còn nhiều thiếu sót. Em rất mong thầy cô có thể giúp đỡ để em hoàn thiện hơn.

## TÀI LIỆU THAM KHẢO

- [1] M. Crabtree, "What is Machine Learning? Definition, Types, Tools & More," 2023. [Online]. Available: <https://www.datacamp.com/blog/what-is-machine-learning>.
- [2] V. H. Tiệp, "Phân nhóm các thuật toán Machine Learning," [Online]. Available: <https://machinelearningcoban.com/2016/12/27/categories/>.
- [3] ibm, "What is deep learning?," [Online]. Available: <https://www.ibm.com/topics/deep-learning>.
- [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 a neural network?," [Online]. Available: <https://www.ibm.com/topics/neural-networks>.
- [6] ibm, "What are convolutional neural networks?," [Online]. Available: <https://www.ibm.com/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>.
- [8] P. Sharma, "Applications of Convolutional Neural Networks(CNN)," 2023. [Online]. Available: <https://www.analyticsvidhya.com/blog/2021/10/applications-of-convolutional-neural-networkscnn/#:~:text=The%20most%20fascinating%20image%20recognition,accuracy%20than%20the%20human%20eye..>
- [9] N. T. Tuấn, Deep Learning Cơ Bản, 2020.
- [10] "Cats vs Dogs : Image Classification using CNN," [Online]. Available: <https://www.kaggle.com/code/ifeoluwaoduwaiye/cats-vs-dogs-image-classification-using-cnn-95/notebook>.