

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 : Lê Duy Quang

Giảng viên hướng dẫn: TS. Ngô Trường Giang

Hải Phòng - 2024

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



XÂY DỰNG MÔ HÌNH HỌC MÁY
CHO BÀI TOÁN PHÂN BIỆT GIỚI TÍNH
DỰA TRÊN KHUÔN MẶT

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

Sinh viên thực hiện : Lê Duy Quang

Giảng viên hướng dẫn: TS. Ngô Trường Giang

Hải Phòng - 2024

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

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên : Lê Duy Quang - **MSV** : 1912111001

Lớp : CT2301M

Ngành : Công nghệ Thông tin

Tên đề tài : Xây dựng mô hình học máy cho bài toán phân biệt giới tính dựa trên khuôn mặt

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp (về lý luận, thực tiễn, các số liệu cần tính toán và các bản vẽ).

Khuôn mặt là một đối tượng trong cơ thể con người và hình ảnh khuôn mặt mang rất nhiều thông tin quan trọng như: tuổi tác, giới tính, trạng thái cảm xúc, dân tộc,... Trong đó, việc xác định giới tính là hết sức quan trọng đối với việc xây dựng hệ thống AI có chức năng giao tiếp.

Đề tài này sẽ tìm hiểu bài toán phân biệt giới tính người dựa trên đặc trưng khuôn mặt sử dụng kiến trúc mạng nơ-ron tích chập sâu với các kỹ thuật liên quan. Nội dung bao gồm:

- Tìm hiểu tổng quan về xử lý ảnh và bài toán phân loại ảnh
- Tìm hiểu mô hình học sâu với một số kiến trúc mạng phổ biến.
- Ứng dụng mô hình học sâu vào bài toán phân loại ảnh với các bước:
 - Chuẩn bị dữ liệu ảnh khuôn mặt (chỉ phần khuôn mặt).
 - Xây dựng mô hình mạng
 - Huấn luyện mô hình
 - Kiểm thử mô hình và đánh giá.
- Báo cáo và trình bày kết quả

2. Các số liệu cần thiết để tính toán.

.....
.....
.....
.....

3. Địa điểm thực tập tốt nghiệp.

CÁC CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Họ và tên : Ngô Trường Giang

Học hàm, học vị : Tiến sĩ

Cơ quan công tác : Trường Đại học Thủy lợi

Nội dung hướng dẫn:

- Tìm hiểu tổng quan bài toán phân loại ảnh.
- Tìm hiểu mô hình học sâu với một số kiến trúc mạng phổ biến.
- Ứng dụng mô hình học sâu vào bài toán phân loại ảnh.
- Cài đặt mô hình mạng nơ-ron nhân chập áp dụng cho phân loại ảnh với các bước:
 - Chuẩn bị dữ liệu ảnh khuôn mặt.
 - Xây dựng mô hình mạng
 - Huấn luyện mô hình
 - Kiểm thử mô hình và đánh giá.

Đề tài tốt nghiệp được giao ngày tháng năm 2024

Yêu cầu phải hoàn thành xong trước ngày tháng năm 2024

Đã nhận nhiệm vụ ĐTTN

Sinh viên

Lê Duy Quang

Đã giao nhiệm vụ ĐTTN

Giảng viên hướng dẫn



Ngô Trường Giang

Hải Phòng, ngày tháng 11 năm 2024

TRƯỞNG KHOA

Cộng hòa Xã hội Chủ nghĩa Việt Nam
Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN TỐT NGHIỆP

Họ và tên giảng viên : Ngô Trường Giang

Đơn vị công tác : Trường Đại học Thủy lợi

Họ và tên sinh viên : Lê Duy Quang

Chuyên ngành : Công nghệ Thông tin

Nội dung hướng dẫn : Toàn bộ đề tài

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp

- Sinh viên có thái độ làm việc nghiêm túc tích cực, chủ động tìm đọc các tài liệu liên quan tới đề tài.
- Chấp hành nghiêm túc kế hoạch, tiến độ đề ra.

2. Đánh giá chất lượng của đề án/khóa luận (so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ.T.T.N, trên các mặt lý luận, thực tiễn, tính toán số liệu...)

Đề tài thuộc lĩnh vực mới và khó đối với sinh viên, nhưng sinh viên đã rất cố gắng để hoàn thành yêu cầu đề ra.

- Về mặt lý thuyết, sinh viên đã tìm hiểu được: Tổng quan về mạng nơron tích chập; bài toán phân loại ảnh; tìm hiểu mạng CNN theo kiến trúc cơ bản để cài đặt thử nghiệm.
- Về mặt thực nghiệm: Sinh viên đã cài đặt mô hình CNN cho bài toán phân loại ảnh (phát hiện giới tính thông qua khuôn mặt, với giới hạn ảnh chỉ có phần khuôn mặt và mỗi ảnh chỉ có 1 khuôn mặt). Mô hình cài đặt trên môi trường googlecolab với ngôn ngữ python: Chuẩn bị dữ liệu, xây dựng và huấn luyện mạng, kiểm thử mô hình. Tuy nhiên phần thử nghiệm mới chỉ dừng lại ở mức độ thử nghiệm đơn giản.
- Báo cáo trình bày sáng sủa, bố cục hợp lý, đáp ứng được yêu cầu đề ra.

3. Ý kiến của giảng viên hướng dẫn tốt nghiệp

Được bảo vệ ☒ Không được bảo vệ ☐ Điểm hướng dẫn ☐

Hải Phòng, ngày 11 tháng 11 năm 2024

Giảng viên hướng dẫn



Ngô Trường Giang

Cộng hòa xã hội chủ nghĩa Việt Nam
Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN CHẤM PHẢN BIỆN

Họ và tên giảng viên

Đơn vị công tác:.....

Họ và tên sinh viên:Chuyên ngành:.....

Đề tài tốt nghiệp:

.....

1. Phần nhận xét của giảng viên chấm phản biện

.....

.....

.....

.....

2. Những mặt còn hạn chế

.....

.....

.....

.....

3. Ý kiến của giảng viên chấm phản biện

Được bảo vệ ☐ Không được bảo vệ ☐ Điểm phản biện ☐

Hải Phòng, ngày.....tháng.....năm 2024

Giảng viên chấm phản biện
(ký và ghi rõ họ tên)

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 Quản lý và Công nghệ Hải Phòng, 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 nhiều 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 đào 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 khoá học và đồ án theo quy định.

Em xin chân thành cảm ơn !

LỜI CẢM ƠN	8
MỞ ĐẦU.....	11
CHƯƠNG 1: Tổng quan về xử lý ảnh và học máy	14
1.1 Giới thiệu về xử lý ảnh.....	14
1.1.1 Xử lý ảnh là gì?	14
1.1.2 Định nghĩa và mục tiêu.....	14
1.1.3 Các bước trong xử lý ảnh	15
Ứng dụng của xử lý ảnh	15
1.2 Một số kỹ thuật xử lý ảnh.....	16
1.2.1 Lọc ảnh (Filtering).....	16
1.2.2 Biến đổi hình học (Geometric Transformation).....	17
1.2.3 Phân đoạn ảnh (Image Segmentation).....	18
1.2.4 Chuyển đổi không gian màu (Color Space Transformation)	18
1.2.5 Nhận diện đối tượng (Object Detection).....	19
1.2.6 Trích xuất đặc trưng (Feature Extraction).....	19
1.3 Học Máy	20
1.3.1 Học có giám sát (Supervised Learning)	20
1.3.2 Học không giám sát (Unsupervised Learning).....	22
1.3.3 Học tăng cường (Reinforcement Learning)	24
CHƯƠNG 2: Mạng nơ-ron tích chập.....	26
2.1 Giới thiệu Học máy (Machine learning)	26
2.2 Giới thiệu học sâu (Deep Learning).....	27
2.2.2 Nguyên tắc hoạt động.....	28
2.2.3 Ứng dụng của học sâu trong thực tế.....	29
2.2.4 Ưu điểm và nhược điểm của học sâu	31
2.3 Bài toán phân lớp dữ liệu	32
2.4 Mạng nơ-ron.....	33
2.4.1 Định nghĩa	33
2.4.2 Nguyên tắc hoạt động.....	34
2.5 Mạng nơ-ron tích chập (Convolutional Neural Network - CNN).....	35
2.5.1 Định nghĩa	35
2.5.2 Nguyên tắc hoạt động.....	35
2.5.3 Ứng dụng của CNN trong thực tế	37
2.5.4 Ưu điểm và nhược điểm của mạng nơ-ron tích chập	38

CHƯƠNG 3: Ứng dụng CNN cho bài toán phân loại ảnh.....	40
3.1 Môi trường cài đặt	40
3.1.1 Google Colab	40
3.1.2 Cài đặt môi trường Google Colab	41
3.1.3 Các thư viện được sử dụng.....	41
3.2 Python.....	42
3.2.1 Định nghĩa	42
3.2.2 Tại sao Python lại được sử dụng nhiều trong học máy?	43
3.3 Thu thập dữ liệu	44
3.4 Tiền xử lý dữ liệu	45
3.5 Xây dựng mô hình.....	46
3.5.1 Xây dựng lớp cho mô hình.....	46
3.5.1 Biên dịch và huấn luyện mô hình.....	48
3.6 Đánh giá mô hình	48
3.7 Một số kết quả thử nghiệm.....	49
3.7.1 Xây dựng hàm dự đoán 1 ảnh.....	49
3.7.1 Thử nghiệm với tập ảnh test	49
3.7.2 Thử nghiệm với tập ảnh huấn luyện.....	52
KẾT LUẬN	57
TÀI LIỆU THAM KHẢO.....	58

MỞ ĐẦU

Trong bối cảnh cuộc cách mạng công nghệ 4.0, hình ảnh và video đã trở thành một phần thiết yếu trong giao tiếp và tương tác giữa con người với máy móc. Sự phát triển nhanh chóng của công nghệ nhận diện hình ảnh đã mở ra nhiều cơ hội mới trong các lĩnh vực như an ninh, truyền thông, và thương mại điện tử. Trong số các ứng dụng nổi bật, phân biệt giới tính dựa trên khuôn mặt là một chủ đề thu hút sự quan tâm của nhiều nhà nghiên cứu và doanh nghiệp. Bằng cách sử dụng các phương pháp xử lý ảnh và học máy, chúng ta có thể xây dựng những hệ thống tự động có khả năng phân loại giới tính của người dùng chỉ bằng cách phân tích hình ảnh khuôn mặt.

Phân biệt giới tính từ khuôn mặt không chỉ đơn thuần là một bài toán nhận diện; nó còn liên quan đến nhiều yếu tố như ngữ cảnh văn hóa, tâm lý học, và các đặc điểm sinh học. Nghiên cứu trong lĩnh vực này giúp làm sáng tỏ cách mà con người tương tác với nhau, cũng như cách mà công nghệ có thể tác động đến sự hiểu biết về giới tính. Bên cạnh đó, khả năng phân biệt giới tính chính xác có thể hỗ trợ trong việc phát triển các ứng dụng cá nhân hóa, như quảng cáo, nội dung truyền thông, và cải thiện trải nghiệm người dùng trong nhiều dịch vụ trực tuyến.

Tuy nhiên, việc phân biệt giới tính từ khuôn mặt gặp phải nhiều thách thức đáng kể. Đầu tiên, sự đa dạng về đặc điểm khuôn mặt giữa các cá nhân có thể gây khó khăn cho mô hình phân tích. Thực tế, khuôn mặt của một người có thể thay đổi theo nhiều yếu tố như tuổi tác, sắc tộc, và thậm chí cả cảm xúc. Hơn nữa, ánh sáng, góc chụp, và chất lượng hình ảnh cũng ảnh hưởng lớn đến hiệu suất của các mô hình nhận diện. Những yếu tố này tạo ra những thách thức không nhỏ trong việc phát triển một hệ thống chính xác và đáng tin cậy.

Đồng thời, sự phát triển của công nghệ học sâu đã mở ra một kỷ nguyên mới cho việc xử lý ảnh và nhận diện khuôn mặt. Các mạng nơ-ron tích chập (CNN) đã chứng minh hiệu quả vượt trội trong việc trích xuất đặc trưng từ hình ảnh, và nhiều nghiên cứu gần đây đã áp dụng thành công các kỹ thuật này để cải thiện độ chính xác của việc phân loại giới tính. Tuy nhiên, còn nhiều cơ hội để tối

ưu hóa các mô hình này, bao gồm việc áp dụng các phương pháp tăng cường dữ liệu, cải thiện kiến trúc mạng, và nghiên cứu các thuật toán học sâu mới.

Với tất cả những lý do này, đề tài "Xây dựng mô hình học máy cho bài toán phân biệt giới tính dựa trên khuôn mặt" không chỉ mang lại cơ hội nghiên cứu sâu sắc về công nghệ mà còn mở ra nhiều hướng đi mới cho các ứng dụng trong tương lai. Việc tìm hiểu và phát triển mô hình phân biệt giới tính sẽ không chỉ nâng cao khả năng nhận diện của các hệ thống mà còn góp phần vào sự hiểu biết sâu hơn về các yếu tố xã hội và tâm lý liên quan đến giới tính trong bối cảnh công nghệ hiện đại. Những kiến thức và kết quả từ nghiên cứu này sẽ đóng góp tích cực vào việc cải thiện các sản phẩm công nghệ, đồng thời tạo ra các giá trị xã hội thiết thực.

1. Lý do chọn đề tài.

Việc phân biệt giới tính từ khuôn mặt không chỉ đơn thuần là một bài toán nhận diện mà còn thể hiện sự kết hợp giữa công nghệ học máy và xử lý ảnh. Lý do chọn đề tài này bao gồm:

Tính ứng dụng cao: Với sự gia tăng của các hệ thống nhận diện khuôn mặt trong cuộc sống hàng ngày, từ điện thoại thông minh đến hệ thống an ninh, khả năng phân biệt giới tính có thể hỗ trợ nhiều dịch vụ cá nhân hóa và bảo mật.

Thách thức kỹ thuật: Mặc dù có nhiều tiến bộ trong lĩnh vực học máy và xử lý ảnh, việc phân biệt giới tính từ khuôn mặt vẫn gặp phải nhiều khó khăn, đặc biệt trong các tình huống ánh sáng không đồng đều, góc chụp khác nhau, và các yếu tố môi trường khác. Đề tài này sẽ giúp tìm hiểu sâu hơn về những thách thức này và cách giải quyết chúng.

Tầm quan trọng trong nghiên cứu xã hội: Việc hiểu và phân tích các yếu tố hình ảnh liên quan đến giới tính không chỉ có ý nghĩa trong lĩnh vực công nghệ mà còn góp phần vào các nghiên cứu xã hội, giúp chúng ta hiểu rõ hơn về cách nhận diện và hình ảnh hóa giới tính trong văn hóa hiện đại.

Sự phát triển của công nghệ: Với sự phát triển không ngừng của các phương pháp học sâu (deep learning), đặc biệt là các mạng nơ-ron tích chập (CNN), có nhiều cơ hội để áp dụng những công nghệ mới nhất vào việc giải

quyết bài toán này. Đề tài này sẽ cung cấp cái nhìn tổng quan về cách các mô hình học máy có thể được xây dựng và tối ưu hóa để đạt được hiệu suất cao.

Với những lý do trên, Em đã lựa chọn đề tài "Xây dựng mô hình học máy cho bài toán phân biệt giới tính dựa trên khuôn mặt" không chỉ là một thử thách kỹ thuật mà còn mở ra cơ hội nghiên cứu sâu hơn về ảnh hưởng của công nghệ trong cuộc sống và xã hội hiện đại

Cấu trúc đồ án gồm 3 chương:

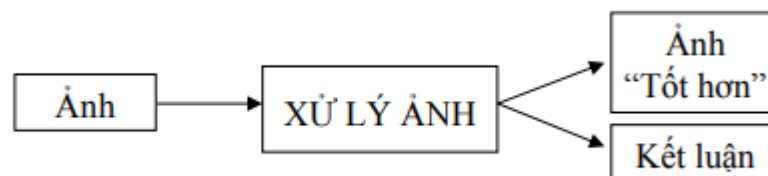
- Chương 1: Tổng quan về xử lý ảnh và học máy
- Chương 2: Mạng nơ-ron tích chập
- Chương 3: Ứng dụng CNN cho phân loại ảnh

CHƯƠNG 1: Tổng quan về xử lý ảnh và học máy

1.1 Giới thiệu về xử lý ảnh

1.1.1 Xử lý ảnh là gì?

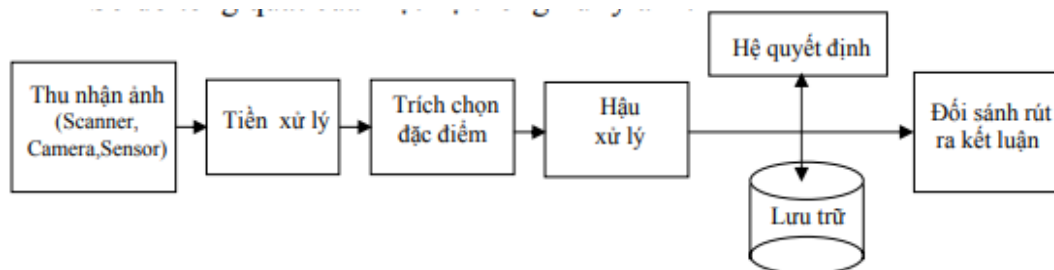
Con người thu nhận thông tin qua các giác quan, trong đó thị giác đóng vai trò quan trọng nhất. Những năm trở lại đây với sự phát triển của phần cứng máy tính, xử lý ảnh và đồ họa đã phát triển một cách mạnh mẽ và có nhiều ứng dụng trong cuộc sống. Xử lý ảnh và đồ họa đóng một vai trò quan trọng trong tương tác người máy. Quá trình xử lý ảnh được xem như là quá trình thao tác ảnh đầu vào nhằm cho ra kết quả mong muốn. Kết quả đầu ra của một quá trình xử lý ảnh có thể là một ảnh “tốt hơn” hoặc một kết luận.



Hình 1-1 Quá trình xử lý ảnh

Ảnh có thể xem là tập hợp các điểm ảnh và mỗi điểm ảnh được xem như là đặc trưng cường độ sáng hay một dấu hiệu nào đó tại một vị trí nào đó của đối tượng trong không gian và nó có thể xem như một hàm n biến $P(c_1, c_2, \dots, c_n)$. Do đó, ảnh trong xử lý ảnh có thể xem như ảnh n chiều.

Sơ đồ tổng quát của một hệ thống xử lý ảnh



Hình 1-2 Các bước cơ bản trong một hệ thống xử lý ảnh

1.1.2 Định nghĩa và mục tiêu

Xử lý ảnh có thể được định nghĩa là một quá trình chuyển đổi các hình ảnh từ dạng số hóa sang thông tin có thể sử dụng được. Mục tiêu chính của xử lý ảnh bao gồm:

Cải thiện chất lượng hình ảnh: Loại bỏ nhiễu, tăng cường độ tương phản, và điều chỉnh độ sáng nhằm tạo ra những hình ảnh rõ ràng và sắc nét hơn.

Trích xuất thông tin: Phân tích hình ảnh để tìm kiếm các đặc điểm, mẫu, hoặc thông tin quan trọng, chẳng hạn như biên giới, hình dạng, và kết cấu.

Tự động hóa các tác vụ: Sử dụng các thuật toán để thực hiện các tác vụ như nhận diện và phân loại hình ảnh mà không cần sự can thiệp của con người.

1.1.3 Các bước trong xử lý ảnh

Quá trình xử lý ảnh thường được chia thành các bước chính sau:

Tiền xử lý: Giai đoạn này bao gồm các kỹ thuật nhằm cải thiện chất lượng hình ảnh ban đầu. Các phương pháp như lọc nhiễu (smoothing), điều chỉnh độ sáng và tương phản, và thay đổi kích thước ảnh thường được thực hiện để chuẩn bị dữ liệu cho các bước tiếp theo.

Phát hiện và phân tích đặc trưng: Sau khi tiền xử lý, các đặc trưng quan trọng sẽ được trích xuất từ hình ảnh. Các thuật toán như phát hiện biên, điểm đặc trưng, và phân khúc hình ảnh sẽ giúp xác định những vùng quan trọng trong ảnh cần được phân tích sâu hơn.

Nhận diện đối tượng: Trong giai đoạn này, các mô hình học máy hoặc học sâu sẽ được áp dụng để nhận diện các đối tượng hoặc phân loại các đặc điểm dựa trên những thông tin đã trích xuất. Điều này rất quan trọng trong bài toán phân biệt giới tính từ khuôn mặt, nơi mà các mô hình cần xác định giới tính dựa trên các đặc trưng khuôn mặt.

Hậu xử lý: Cuối cùng, kết quả nhận được từ các bước trên sẽ được xử lý thêm để tạo ra các thông tin dễ hiểu và có giá trị cho người sử dụng. Điều này có thể bao gồm việc tạo báo cáo, hình ảnh trực quan, hoặc các kết quả phân tích khác.

Ứng dụng của xử lý ảnh

Xử lý ảnh có mặt trong nhiều lĩnh vực khác nhau, từ công nghiệp đến y tế, và cả trong đời sống hàng ngày. Một số ứng dụng tiêu biểu bao gồm:

- Nhận diện khuôn mặt: Sử dụng trong bảo mật, xác thực người dùng, và các hệ thống giám sát.
- Y tế: Phân tích hình ảnh y khoa như X-quang, MRI, và CT để hỗ trợ chẩn đoán bệnh.

- Giao thông: Phân tích hình ảnh từ camera giám sát để phát hiện và phân tích lưu lượng giao thông.
- Nông nghiệp: Sử dụng drone và công nghệ máy bay không người lái để theo dõi tình trạng mùa màng, phát hiện bệnh tật và sâu bệnh.

1.2 Một số kỹ thuật xử lý ảnh

Xử lý ảnh là quá trình thay đổi hoặc cải thiện hình ảnh nhằm đạt được mục tiêu cụ thể, chẳng hạn như cải thiện chất lượng, nhận diện đối tượng, hoặc phân tích các đặc trưng trong hình ảnh. Các kỹ thuật xử lý ảnh có thể bao gồm các bước như làm mịn ảnh, phát hiện biên, phân đoạn ảnh, thay đổi không gian màu, hoặc thậm chí tái tạo ảnh. Dưới đây là một số kỹ thuật xử lý ảnh phổ biến, được sử dụng rộng rãi trong các ứng dụng thực tế.

1.2.1 Lọc ảnh (Filtering)

Lọc ảnh là một kỹ thuật quan trọng trong xử lý ảnh, dùng để làm sạch ảnh, giảm nhiễu, hoặc làm nổi bật các đặc điểm cụ thể trong ảnh. Các loại lọc phổ biến bao gồm:

Lọc trung bình (Mean filtering):

- Mục đích: Làm mượt hình ảnh và loại bỏ nhiễu.
- Nguyên lý: Mỗi pixel trong ảnh được thay thế bằng giá trị trung bình của các pixel lân cận trong một cửa sổ (kernel) nhất định.
- Ứng dụng: Dùng trong các tình huống khi ảnh bị nhiễu hoặc muốn làm giảm chi tiết không cần thiết.

Lọc Gaussian:

- Mục đích: Làm mờ ảnh và giảm nhiễu, nhưng giữ lại biên và các đặc điểm quan trọng.
- Nguyên lý: Áp dụng hàm Gaussian lên mỗi pixel để làm mờ ảnh, trong đó sự ảnh hưởng của mỗi pixel lân cận giảm dần theo khoảng cách.
- Ứng dụng: Dùng trong các tình huống cần làm mờ ảnh mà không làm mất đi quá nhiều chi tiết.

Lọc biên (Edge detection):

- Mục đích: Phát hiện biên và cấu trúc trong ảnh.
- Các thuật toán lọc biên phổ biến:
- Thuật toán Sobel: Dùng để phát hiện biên trong ảnh bằng cách tính độ dốc của giá trị pixel.
- Thuật toán Canny: Phát hiện biên mạnh mẽ hơn bằng cách sử dụng nhiều bước xử lý như giảm nhiễu, tính gradient, và ngưỡng biên.
- Ứng dụng: Phân đoạn ảnh, nhận diện đối tượng, phân tích hình học.

1.2.2 Biến đổi hình học (Geometric Transformation)

Biến đổi hình học trong xử lý ảnh liên quan đến việc thay đổi vị trí, hình dạng hoặc kích thước của ảnh. Các phép biến đổi phổ biến bao gồm:

Phép xoay (Rotation):

- Mục đích: Xoay hình ảnh quanh một điểm cố định (thường là trung tâm) với một góc xác định.
- Ứng dụng: Sử dụng khi ảnh bị quay lệch hoặc cần xoay một ảnh sao cho phù hợp với yêu cầu xử lý.

Phép thu nhỏ (Resizing):

- Mục đích: Thay đổi kích thước của ảnh.
- Ứng dụng: Dùng để thay đổi độ phân giải của ảnh, ví dụ như giảm kích thước ảnh để tiết kiệm băng thông trong các ứng dụng web hoặc truyền thông.

Phép cắt xén (Cropping):

- Mục đích: Lấy một phần của ảnh, cắt bỏ các khu vực không quan trọng.
- Ứng dụng: Dùng để tập trung vào các vùng cụ thể trong ảnh hoặc để làm sạch các vùng không cần thiết.

Phép biến dạng (Warping):

- Mục đích: Thực hiện các biến dạng phi tuyến tính để làm thay đổi hình dạng của ảnh.

- Ứng dụng: Trong các ứng dụng tạo hiệu ứng đặc biệt, tạo ảnh động hoặc trong các mô hình 3D.

1.2.3 Phân đoạn ảnh (Image Segmentation)

Phân đoạn ảnh là quá trình chia ảnh thành các vùng (segment) khác nhau, với mỗi vùng đại diện cho một đối tượng hoặc phần của đối tượng trong ảnh.

Phân đoạn dựa trên ngưỡng (Thresholding):

- Mục đích: Tách đối tượng khỏi nền bằng cách xác định một ngưỡng mức độ sáng tối.
- Ứng dụng: Dùng trong các bài toán nhận diện chữ viết, phân tích ảnh y tế (chẳng hạn phân đoạn ảnh tế bào trong y khoa).

Phân đoạn theo k-means (K-means clustering):

- Mục đích: Sử dụng thuật toán phân cụm k-means để phân chia ảnh thành các nhóm dựa trên màu sắc hoặc đặc trưng của các pixel.
- Ứng dụng: Phân đoạn ảnh trong các bài toán nhận diện hoặc phân loại đối tượng.

Phân đoạn theo phương pháp vùng (Region-based segmentation):

- Mục đích: Phân đoạn ảnh dựa trên sự tương đồng của các vùng trong ảnh.
- Ứng dụng: Dùng trong nhận diện đối tượng, ví dụ: phân đoạn các vùng mạch máu trong ảnh y tế.

1.2.4 Chuyển đổi không gian màu (Color Space Transformation)

Chuyển đổi không gian màu là kỹ thuật thay đổi không gian màu của ảnh để thuận tiện hơn cho các phép toán xử lý tiếp theo.

Chuyển đổi từ RGB sang HSV:

- Mục đích: Chuyển không gian màu từ RGB sang HSV (Hue, Saturation, Value) giúp dễ dàng hơn trong việc xử lý các tính chất màu sắc như độ sáng và độ bão hòa.

- Ứng dụng: Dùng trong các bài toán nhận diện màu sắc hoặc phân loại ảnh.

Chuyển đổi sang không gian màu YCbCr:

- Mục đích: Dùng trong các ứng dụng xử lý ảnh video hoặc nén ảnh (như JPEG).
- Ứng dụng: Tối ưu hóa việc nén và truyền tải ảnh, vì không gian màu YCbCr dễ dàng tách riêng độ sáng và độ màu, giúp nén hiệu quả.

1.2.5 Nhận diện đối tượng (Object Detection)

Nhận diện đối tượng là kỹ thuật trong xử lý ảnh dùng để xác định các đối tượng hoặc mẫu có trong ảnh.

Mạng nơ-ron tích chập (Convolutional Neural Networks - CNN):

- Mục đích: CNN là một loại mạng học sâu đặc biệt hiệu quả trong việc nhận diện các đối tượng trong ảnh.
- Ứng dụng: Nhận diện khuôn mặt, nhận diện phương tiện giao thông, phát hiện các vật thể trong ảnh hoặc video.

HOG (Histogram of Oriented Gradients):

- Mục đích: Trích xuất các đặc trưng hình học từ ảnh bằng cách sử dụng độ dốc (gradient) của các pixel trong các hướng khác nhau.
- Ứng dụng: Phát hiện người hoặc các đối tượng trong ảnh.

YOLO (You Only Look Once):

- Mục đích: Là thuật toán nhận diện đối tượng nhanh chóng và chính xác, có thể phát hiện nhiều đối tượng trong một lần duyệt ảnh.
- Ứng dụng: Phát hiện đối tượng trong thời gian thực, như nhận diện trong hệ thống an ninh, xe tự lái.

1.2.6 Trích xuất đặc trưng (Feature Extraction)

Trích xuất đặc trưng là quá trình tìm kiếm các đặc trưng quan trọng từ ảnh để sử dụng trong các bài toán học máy.

SIFT (Scale-Invariant Feature Transform):

- Mục đích: Tìm kiếm các đặc trưng trong ảnh mà không phụ thuộc vào biến đổi tỷ lệ và góc quay.
- Ứng dụng: Nhận diện và so sánh hình ảnh, dùng trong các hệ thống tìm kiếm ảnh, hoặc trong các ứng dụng thực tế như AR (thực tế tăng cường).

SURF (Speeded-Up Robust Features):

- Mục đích: Phát hiện các đặc trưng trong ảnh với tốc độ nhanh hơn SIFT.
- Ứng dụng: Các ứng dụng nhận diện đối tượng, phân tích ảnh.

1.3 Học Máy

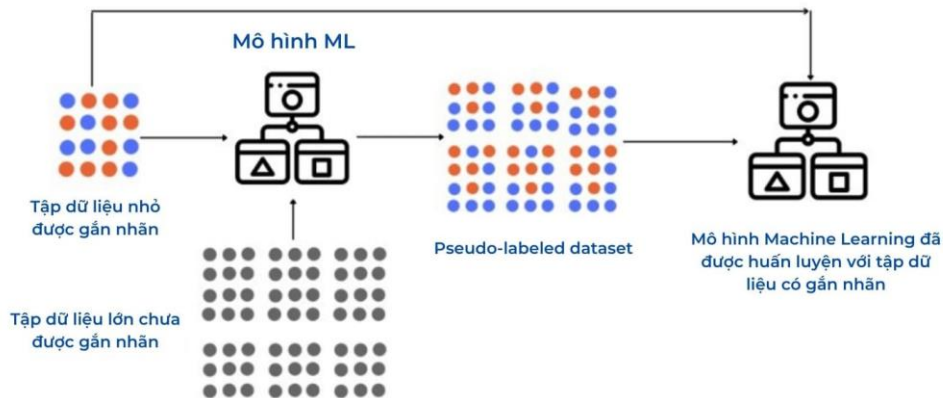
Học máy (Machine Learning - ML) là một nhánh con của trí tuệ nhân tạo (AI), nơi máy tính có thể học từ dữ liệu và cải thiện hiệu suất của mình mà không cần phải lập trình thủ công cho từng nhiệm vụ cụ thể. Học máy đã và đang đóng vai trò quan trọng trong việc giải quyết các bài toán phức tạp, bao gồm nhận diện hình ảnh, phân tích ngữ nghĩa, dự đoán chuỗi thời gian, tự động hóa quy trình và nhiều lĩnh vực khác.

Học máy có thể được chia thành ba phương pháp chính, mỗi phương pháp có những ứng dụng và kỹ thuật riêng biệt.

1.3.1 Học có giám sát (Supervised Learning)

Học có giám sát là phương pháp học mà trong đó một mô hình học từ một tập dữ liệu đã được gán nhãn (labeled data). Dữ liệu huấn luyện trong học có giám sát bao gồm các cặp dữ liệu đầu vào và đầu ra (hoặc nhãn). Mục tiêu của học có giám sát là tìm ra một hàm ánh xạ từ đầu vào đến đầu ra sao cho khi nhận đầu vào mới, mô hình có thể dự đoán đúng nhãn đầu ra.

Học máy bán giám sát



Hình 1-3 Học máy bán giám sát

Các thuật toán học có giám sát phổ biến:

- **Hồi quy tuyến tính (Linear Regression):** Đây là một trong những thuật toán đơn giản nhất trong học máy, được sử dụng chủ yếu cho các bài toán dự đoán giá trị liên tục. Mô hình này học cách vẽ một đường thẳng phù hợp nhất (best-fit line) để dự đoán giá trị của đầu ra dựa trên giá trị đầu vào.
- **Hồi quy logistic (Logistic Regression):** Là một mô hình phân loại dùng để dự đoán xác suất của một sự kiện nào đó (một nhãn phân loại). Ví dụ, nó có thể dự đoán liệu một email là spam hay không spam dựa trên các đặc trưng của email.
- **Cây quyết định (Decision Trees):** Thuật toán này tạo ra một mô hình phân loại hoặc hồi quy dưới dạng cây phân nhánh. Mỗi nút trong cây quyết định đại diện cho một phép thử, và mỗi nhánh biểu thị kết quả của phép thử đó, dẫn đến một nhãn hoặc giá trị cuối cùng.
- **Máy hỗ trợ véc-tơ (Support Vector Machines - SVM):** SVM là một thuật toán phân loại mạnh mẽ, tìm kiếm một siêu phẳng (hyperplane) tối ưu để phân tách các lớp trong không gian đặc trưng sao cho biên độ phân tách giữa các lớp là lớn nhất. SVM cũng có thể được mở rộng cho các bài toán phân loại không tuyến tính thông qua kỹ thuật nhân (kernel trick).

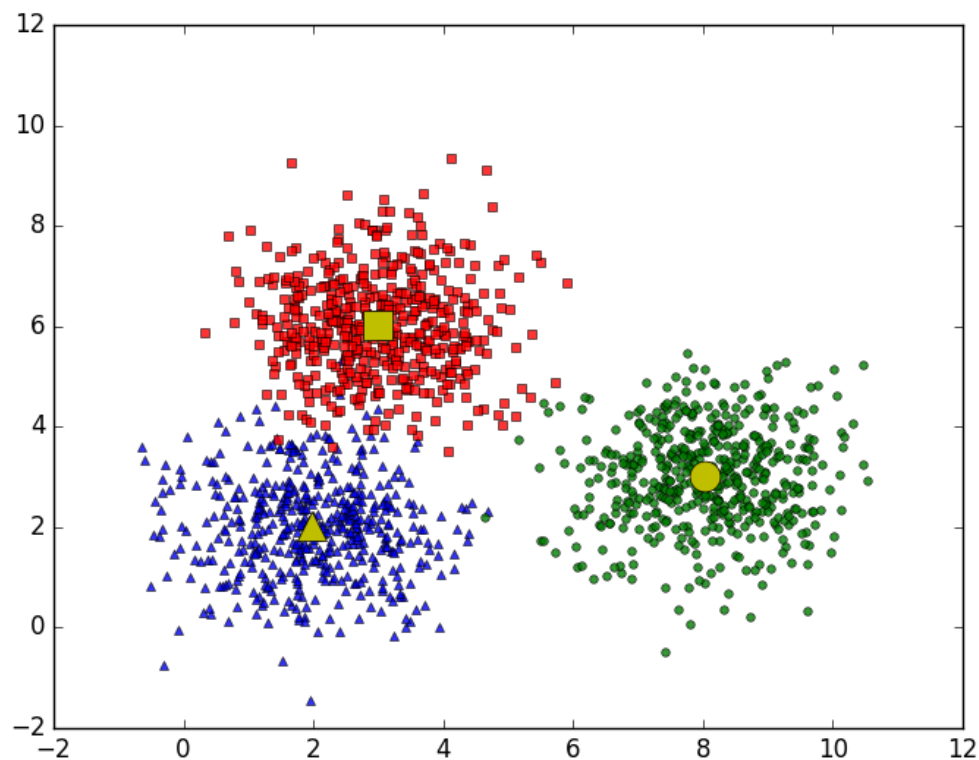
- **K-Nearest Neighbors (K-NN):** Là một thuật toán phân loại đơn giản nhưng mạnh mẽ. Mô hình này dựa vào việc tìm kiếm k điểm dữ liệu gần nhất (neighbors) trong không gian đặc trưng và gán nhãn cho đối tượng dựa trên số lượng nhãn của các điểm gần nhất.

Ứng dụng của học có giám sát:

- **Phân loại văn bản:** Phân loại email thành spam hay không spam, phân loại cảm xúc trong các bình luận, hoặc phân loại tài liệu.
- Nhận diện đối tượng trong ảnh: Nhận diện và phân loại các đối tượng trong ảnh như phân loại chó và mèo, nhận diện khuôn mặt.
- Dự đoán giá trị tài chính: Dự đoán giá cổ phiếu, tỷ giá hối đoái, hoặc các chỉ số tài chính dựa trên dữ liệu lịch sử.

1.3.2 Học không giám sát (Unsupervised Learning)

Khác với học có giám sát, học không giám sát không sử dụng dữ liệu có nhãn. Mục tiêu của học không giám sát là tìm ra cấu trúc ẩn trong dữ liệu mà không cần sự can thiệp của con người về nhãn đầu ra. Các thuật toán học không giám sát thường được sử dụng để phân tích dữ liệu lớn và tìm ra các mẫu hoặc nhóm dữ liệu tương đồng.



Hình 1-4 mô tả phân cụm K-means

Các thuật toán học không giám sát phổ biến:

Phân cụm (Clustering):

- **K-means:** Thuật toán phân cụm phổ biến, chia dữ liệu thành k nhóm sao cho các điểm trong mỗi nhóm có đặc trưng giống nhau nhất.
- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** Một thuật toán phân cụm không phụ thuộc vào số lượng nhóm k, thay vào đó, DBSCAN tìm các cụm dữ liệu có mật độ cao và loại bỏ các điểm nhiễu.
- Giảm chiều dữ liệu (Dimensionality Reduction):
- **PCA (Principal Component Analysis):** Một kỹ thuật để giảm số lượng các biến trong dữ liệu, làm cho quá trình phân tích và mô hình hóa trở nên dễ dàng hơn, đồng thời giúp loại bỏ các đặc trưng không cần thiết.
- **t-SNE (t-Distributed Stochastic Neighbor Embedding):** Một kỹ thuật giảm chiều dữ liệu rất hiệu quả trong việc trực quan hóa dữ liệu có chiều cao (dữ liệu nhiều chiều) xuống các không gian 2D hoặc 3D.

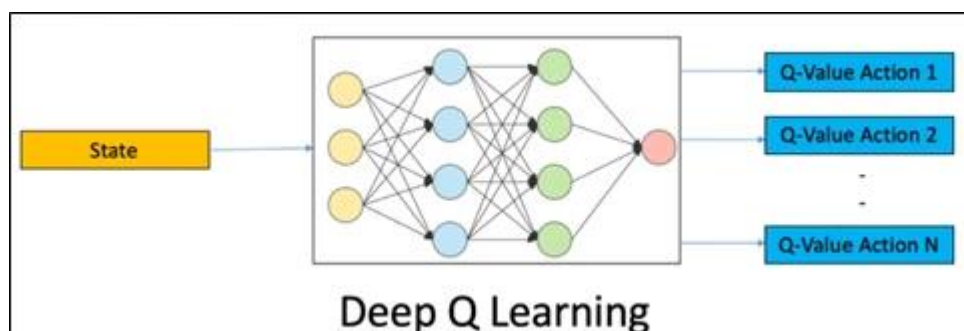
- **Mô hình ẩn (Latent Variable Models):** Các mô hình này giả định rằng dữ liệu quan sát được bị ảnh hưởng bởi các biến ẩn (latent variables) mà không được quan sát trực tiếp, chẳng hạn như mô hình **Mixture Models** (mô hình hỗn hợp).

Ứng dụng của học không giám sát:

- **Phân tích dữ liệu khách hàng:** Tìm các nhóm khách hàng có hành vi tương tự trong lĩnh vực marketing để xây dựng chiến lược tiếp thị mục tiêu.
- **Khám phá dữ liệu:** Phân tích các đặc trưng tiềm ẩn trong dữ liệu y tế, khoa học xã hội, hoặc khảo sát.
- **Khám phá mẫu trong dữ liệu:** Phát hiện các mẫu lạ hoặc bất thường trong các hệ thống giám sát hoặc an ninh mạng (phát hiện xâm nhập).

1.3.3 Học tăng cường (Reinforcement Learning)

Học tăng cường (RL) là một dạng học máy mà trong đó một tác nhân (agent) học cách tối ưu hóa hành động của mình qua việc tương tác với môi trường để tối đa hóa phần thưởng (reward). Tác nhân sẽ học qua các lần thử và sai (trial and error), và trong mỗi bước, nó nhận được phản hồi từ môi trường dưới dạng phần thưởng hoặc hình phạt. Mô hình RL được sử dụng rất hiệu quả trong các bài toán có tính chất quyết định, điều khiển hoặc tự động hóa.



Hình 1-5

Các thành phần chính trong học tăng cường:

- **Tác nhân (Agent):** Là đối tượng thực hiện các hành động trong môi trường.
- **Môi trường (Environment):** Nơi mà tác nhân thực hiện hành động và nhận phản hồi.

- **Phần thưởng (Reward):** Phản hồi mà tác nhân nhận được từ môi trường sau khi thực hiện một hành động. Mục tiêu của tác nhân là tối đa hóa phần thưởng này.
- **Chính sách (Policy):** Là chiến lược mà tác nhân sử dụng để quyết định hành động trong mỗi trạng thái.
- **Hàm giá trị (Value Function):** Đo lường mức độ "tốt" của một trạng thái, tức là giá trị kỳ vọng của phần thưởng mà tác nhân có thể nhận được từ trạng thái đó.

Các thuật toán học tăng cường:

- **Q-learning:** Một thuật toán học tăng cường không cần mô hình, sử dụng bảng Q để lưu trữ giá trị kỳ vọng của mỗi cặp trạng thái-hành động.
- **Deep Q Networks (DQN):** Sự kết hợp của học tăng cường và học sâu, sử dụng mạng nơ-ron để ước lượng hàm Q.
- **Policy Gradient Methods:** Thuật toán học tăng cường này học trực tiếp chính sách của tác nhân mà không cần sử dụng giá trị hàm Q.

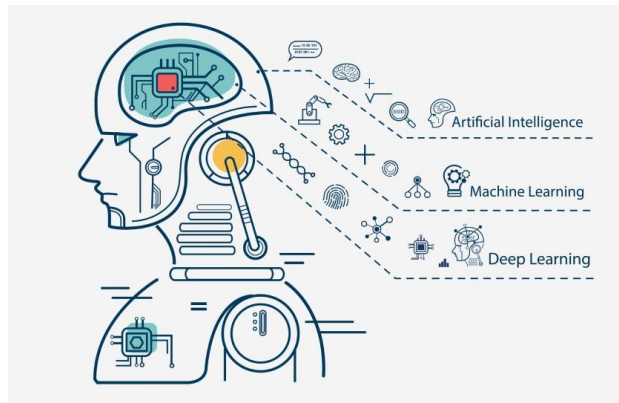
Ứng dụng của học tăng cường:

- **Robot tự động:** Các tác nhân học cách điều khiển robot để hoàn thành nhiệm vụ như di chuyển trong môi trường mà không va chạm.
- **Game và trò chơi:** Học máy được ứng dụng để phát triển các tác nhân có thể chơi game (ví dụ: AlphaGo của Google chơi cờ vây).
- **Xe tự lái:** Học tăng cường giúp xe tự lái học cách điều khiển và đưa ra quyết định trong môi trường giao thông thực tế.

CHƯƠNG 2: Mạng nơ-ron tích chập

2.1 Giới thiệu Học máy (Machine learning)

Học máy là một lĩnh vực con của trí tuệ nhân tạo (AI), tập trung vào việc phát triển các thuật toán và mô hình cho phép máy tính học hỏi từ dữ liệu mà không cần lập trình cụ thể cho từng nhiệm vụ. Thay vì sử dụng các quy tắc lập trình tĩnh, học máy cho phép máy tính phát hiện và học từ mẫu, điều này cực kỳ hữu ích trong các ứng dụng mà dữ liệu có sự biến đổi lớn hoặc không thể dự đoán chính xác.



Hình 2-1 Machine Learning

Máy học (ML - Machine learning) được biết đến là một phần quan trọng trong ngành [trí tuệ nhân tạo AI](#) (Artificial Intelligence) và khoa học máy tính, tập trung vào việc thu thập, phân tích, sử dụng dữ liệu và thuật toán để bắt chước cách con người học, dần dần cải thiện độ chính xác của nó.

Máy học đóng vai trò quan trọng trong nhiều ngành nghề, lĩnh vực. Chính vì vậy, công nghệ này đang được nghiên cứu sâu, phát triển rộng với nhiều ứng dụng hữu ích trong thực tiễn đời sống.

Vậy công nghệ máy học (Machine learning) là gì? Ứng dụng máy học mang lại những lợi ích gì trong đời sống, hãy cùng tham khảo trong nội dung sau đây.

Học máy có thể được chia thành ba loại chính:

- Học có giám sát (Supervised Learning): Trong mô hình này, dữ liệu huấn luyện đã được gán nhãn, và mục tiêu là xây dựng một mô hình có thể dự đoán nhãn cho dữ liệu chưa thấy. Ví dụ, trong bài toán phân biệt giới tính, tập dữ liệu huấn luyện có thể bao gồm hình ảnh khuôn mặt với nhãn "nam" hoặc "nữ".

- Học không giám sát (Unsupervised Learning): Dữ liệu trong trường hợp này không có nhãn. Mô hình phải tự tìm ra các cấu trúc, mẫu hoặc nhóm trong dữ liệu. Các kỹ thuật như phân cụm (clustering) thường được áp dụng trong tình huống này.
- Học tăng cường (Reinforcement Learning): Ở đây, một tác nhân (agent) học cách hành động trong một môi trường để tối đa hóa một hàm thưởng. Các thuật toán này thường được sử dụng trong các bài toán như trò chơi, robot tự hành, và nhiều ứng dụng phức tạp khác.

Học máy đã được áp dụng thành công trong nhiều lĩnh vực, từ nhận diện hình ảnh và giọng nói cho đến dự đoán thị trường tài chính và phân tích y tế.

2.2 Giới thiệu học sâu (Deep Learning)

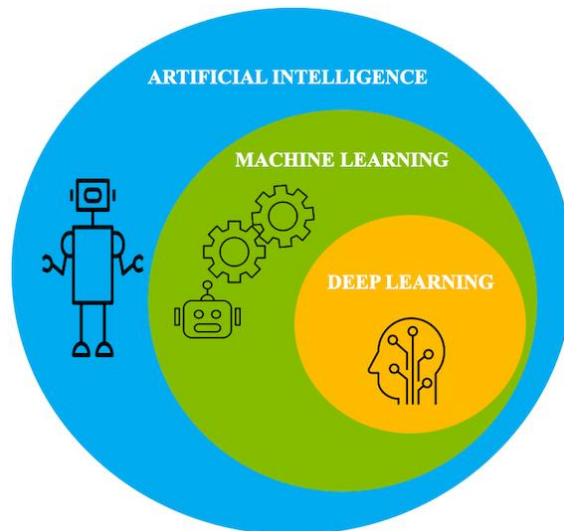
2.2.1.1 Định nghĩa

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.

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.



Hình2-2 Deep learning

2.2.2 Nguyên tắc hoạt động

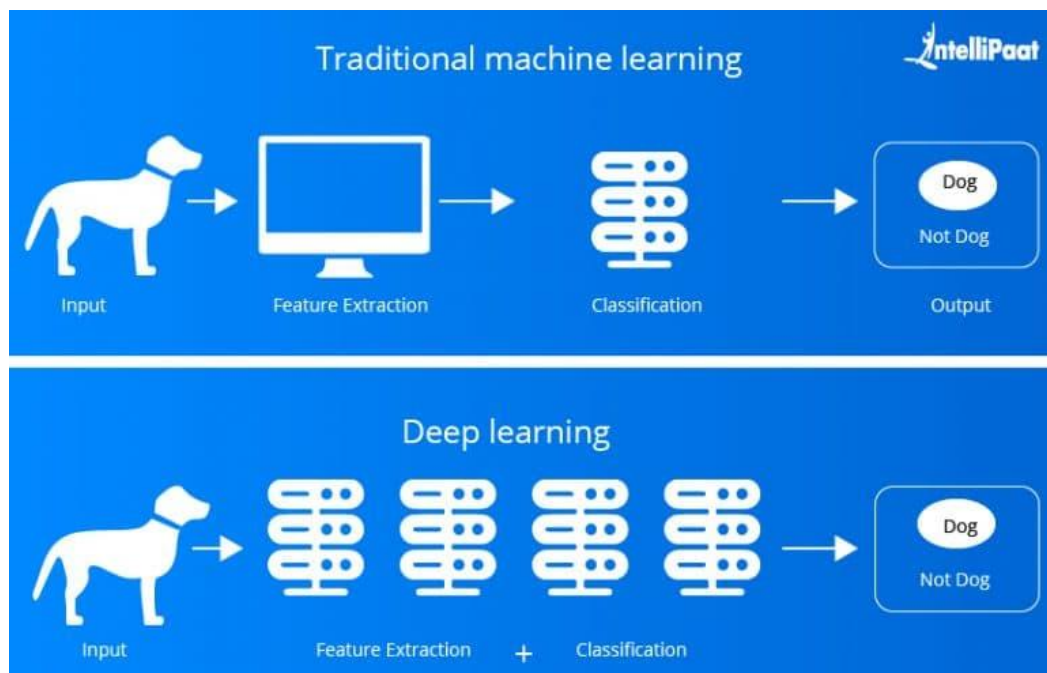
Mạng lưới thần kinh học sâu hoặc mạng lưới thần kinh nhân tạo cố gắng bắt chước bộ não con người thông qua sự kết hợp của dữ liệu đầu vào, trọng số và độ lệch. Các phần tử này phối hợp với nhau để nhận dạng, phân loại và mô tả chính xác các đối tượng trong dữ liệu.

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.

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.
- 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 2-3 Machine Learning vs Deep Learning

2.2.3 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:

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.

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.

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.

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.

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.

2.2.4 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:

- 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.
- 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).
- 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.
- 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.
- 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:

- 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.
- 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.

- 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.
- 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.
- 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.

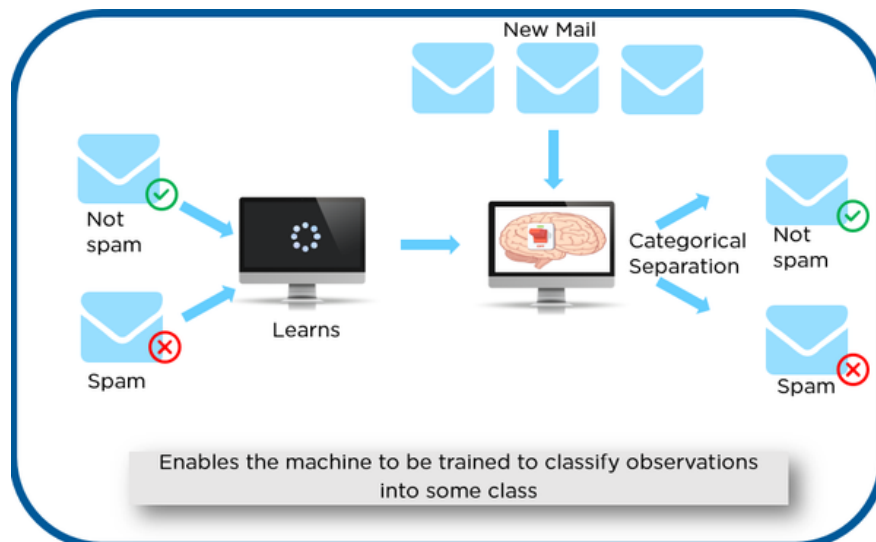
2.3 Bài toán phân lớp dữ liệu

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 2-4 Email Classification

2.4 Mạng nơ-ron

2.4.1 Định nghĩa

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.

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.

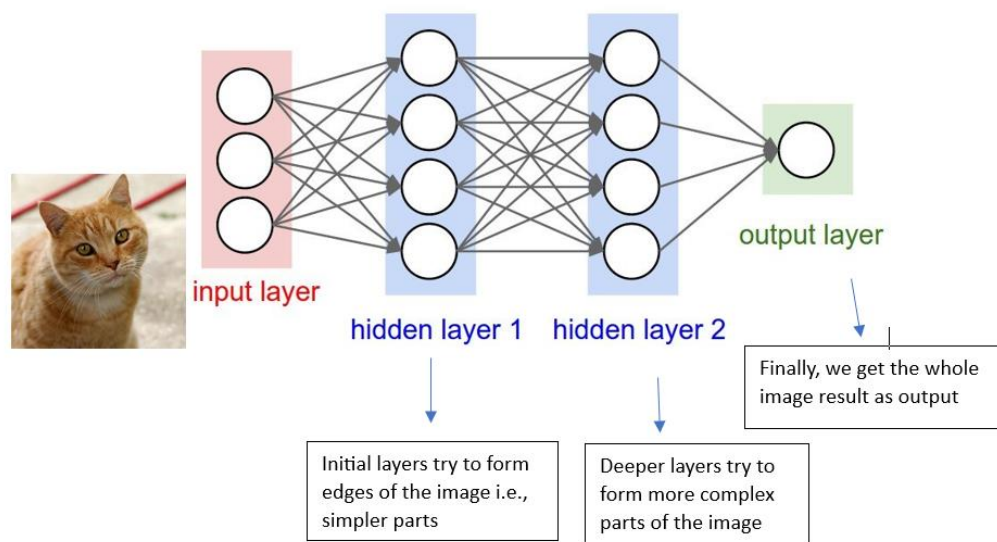
Mạng nơ-ron có khả năng học và tự điều chỉnh để thích nghi với dữ liệu, và chúng được sử dụng rộng rãi trong Machine Learning và Deep Learning để giải quyết các bài toán như nhận dạng hình ảnh, dịch thuật tự động, dự đoán dữ liệu, và nhiều ứng dụng khác.

2.4.2 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:

- 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.
- 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.
- 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.
- 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.
- 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.
- 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 2-5 Mạng nơ-ron

2.5 Mạng nơ-ron tích chập (Convolutional Neural Network - CNN)

2.5.1 Định nghĩa

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.

2.5.2 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.

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.

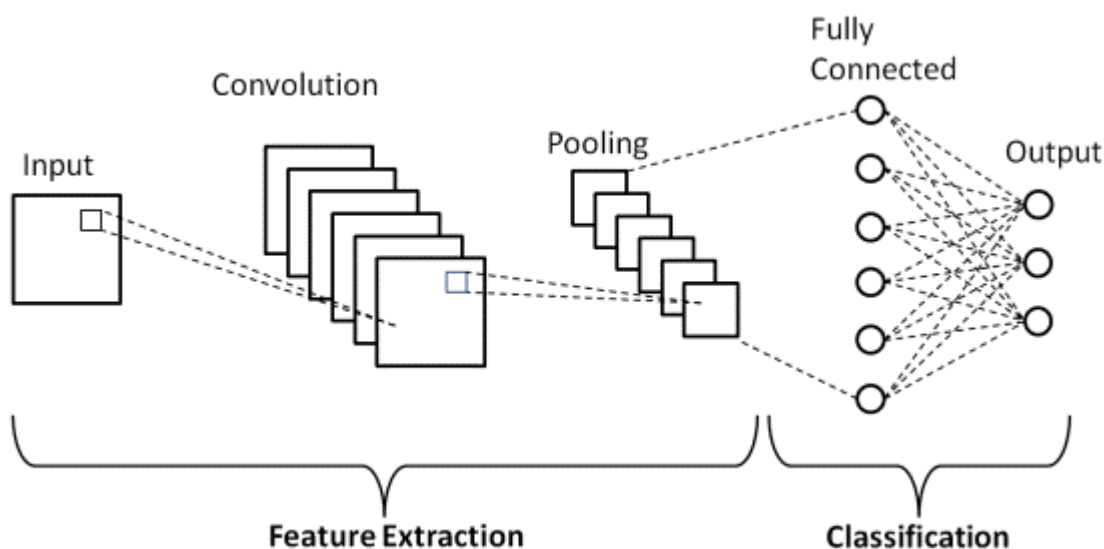
Nguyên tắc hoạt động của mạng tích chập (CNN) có thể được phân thành các bước chính như sau:

- **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.
- **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.
- **Lớp 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 2-6 Ví dụ phân biệt ảnh

2.5.3 Ứng dụng của CNN trong thực tế

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

- **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

2.5.4 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:

- 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.
- 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.
- 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.
- 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.

- 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

- **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.
- **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.
- **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.
- **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.
- **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 CNN cho bài toán phân loại ảnh

3.1 Môi trường cài đặt

3.1.1 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.



Hình 3-1 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 Cài đặt 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.

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 Python

3.2.1 Định nghĩa

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 Python

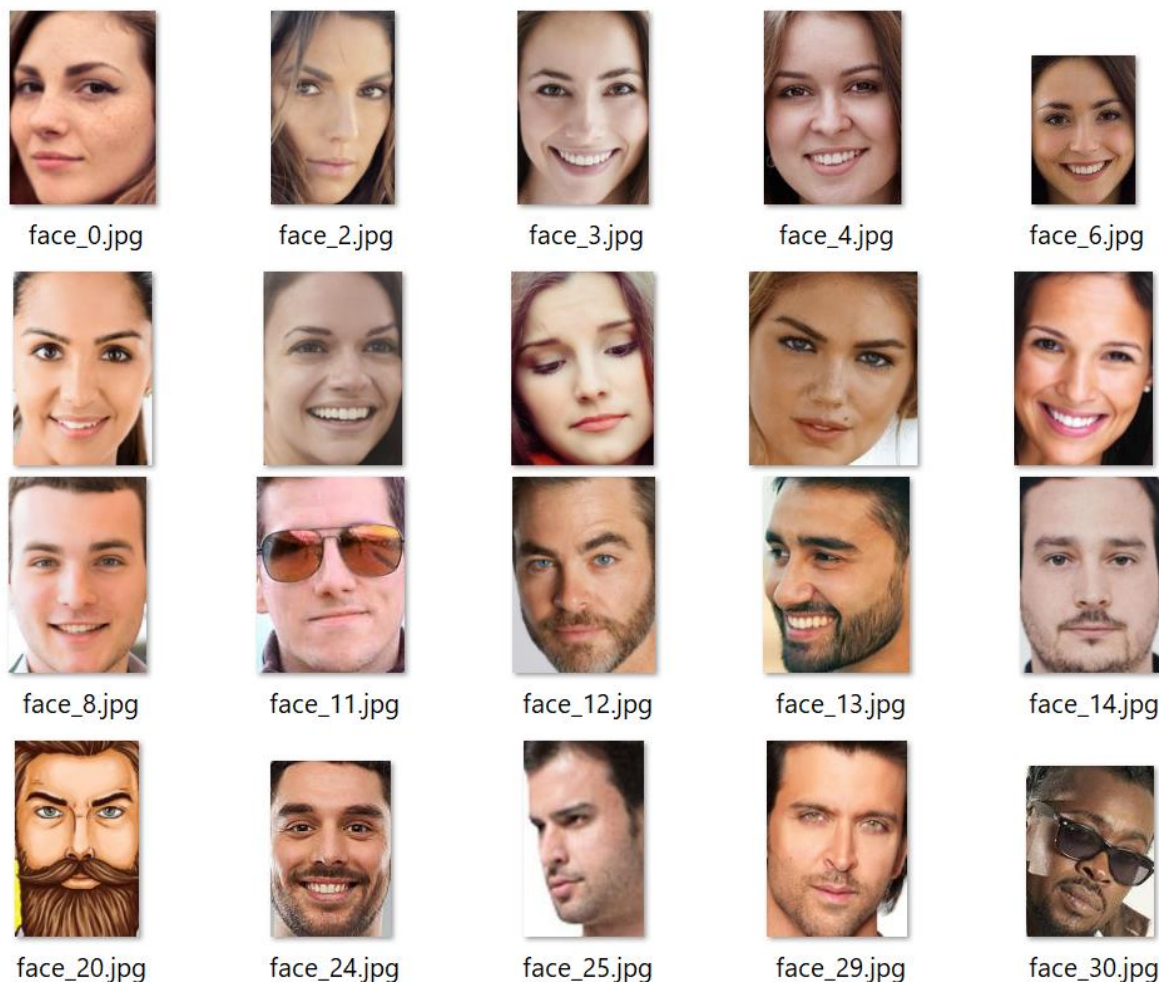
3.2.2 Tại sao Python lại được sử dụng nhiều trong học máy?

Python trong lĩnh vực Machine Learning vì có một số lý do sau:

- 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.
- 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.
- 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.
- 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.
- 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 “male-vs-female” được lấy từ trên Kaggle bao gồm 2907 bức ảnh chia thành 2 tập là tập huấn luyện và tập kiểm thử, trong đó 1528 bức ảnh là ảnh nam 1379 bức ảnh là ảnh nữ. Tập dữ liệu huấn luyện được tổ chức riêng trong 2 thư mục: `train_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`. Dưới đây là một số ảnh trong tập huấn luyện.



Hình 3-3 Một số ảnh mẫu trong tập huấn luyện

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 = 64 # Cỡ của ảnh
image_channel = 3 # Số kênh màu (RGB)
batch_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/ male-vs-female/data/Train/',
                                             class_mode='categorical',
                                             target_size = (image_size,image_size),
                                             batch_size = bat_size,
                                             )

val_gen = test_datagen.flow_from_directory('/content/drive/MyDrive/ male-vs-female/data/Val/',
                                           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 lớp cho mô hình

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

- Tầng đầu vào bao gồm lớp tích chập Conv2D với 32 bộ lọc và hàm kích hoạt relu.
- Mô hình chứa 1 thêm tầng ẩn chứa lớp tích chập với các bộ lọc tăng dần và hàm kích hoạt relu.
- Mỗi tầng tích chập chứa Batch Normalization (giảm thiểu overfitting), Max pooling và Dropout (ngăn chặn overfitting).
- Các lớp được kết nối đầy đủ bao gồm lớp Flatten, lớp Dense với 512 đơn vị (số nơ-ron) và lớp Dropout.
- Lớp đầu ra là lớp Dense với 2 đơn vị (2 lớp đầu ra) 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: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 32)	896
batch_normalization (BatchNormalization)	(None, 62, 62, 32)	128
max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	0
dropout (Dropout)	(None, 31, 31, 32)	0
conv2d_1 (Conv2D)	(None, 29, 29, 64)	18,496
batch_normalization_1 (BatchNormalization)	(None, 29, 29, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 64)	0
dropout_1 (Dropout)	(None, 14, 14, 64)	0
flatten (Flatten)	(None, 12544)	0
dense (Dense)	(None, 512)	6,423,040
batch_normalization_2 (BatchNormalization)	(None, 512)	2,048
dropout_2 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 2)	1,026

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()
```

3.5.1 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
Male_female = model.fit(train_gen,
                        validation_data = val_gen,
                        callbacks=[learning_rate_reduction],
                        epochs = 10,
                        # steps_per_epoch = len(train_generator),
                        # validation_steps = len(val_generaotor),
                        )
# Lưu mô hình
model.save("/content/drive/MyDrive/male-vs-female/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 (male hoặc female)

#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 = 'female'

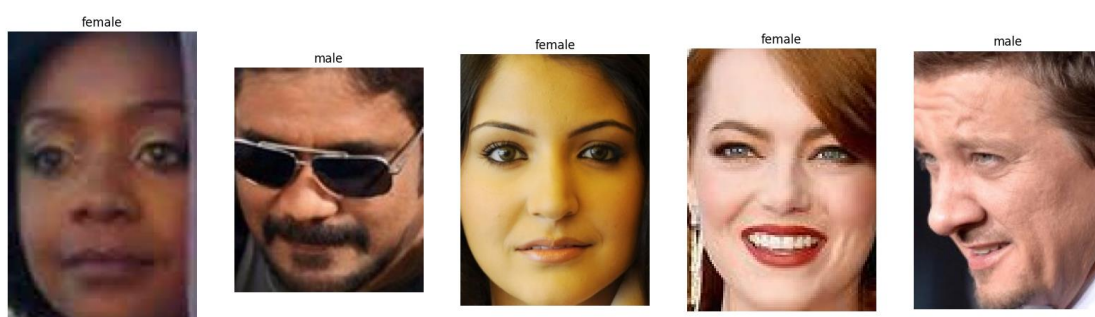
else:

pred='male'

return [img, pred]

3.7.1 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:



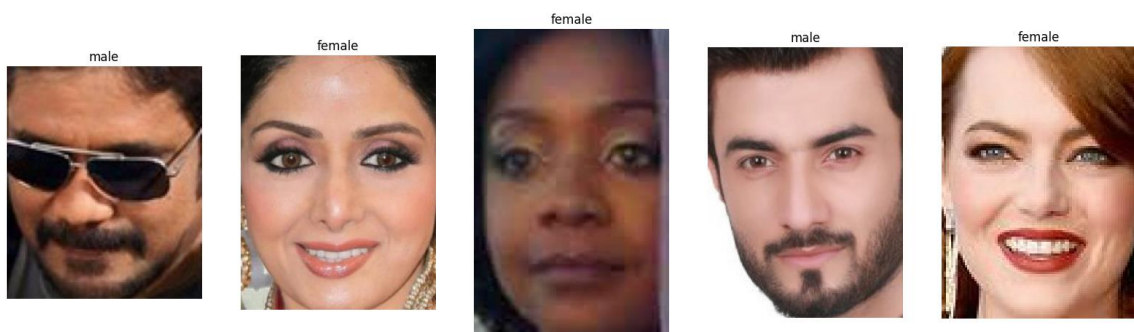
Hình 3-1 Kết quả test_1 trên tập test



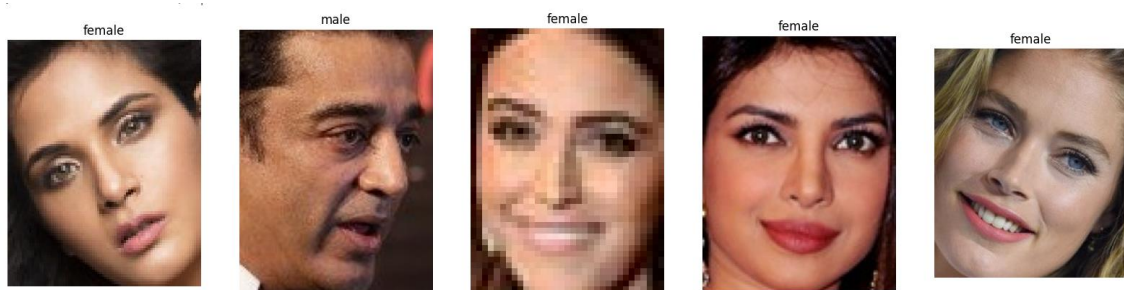
Hình 3-6 Kết quả test_2 trên tập test



Hình 3-7 Kết quả test_3 trên tập test



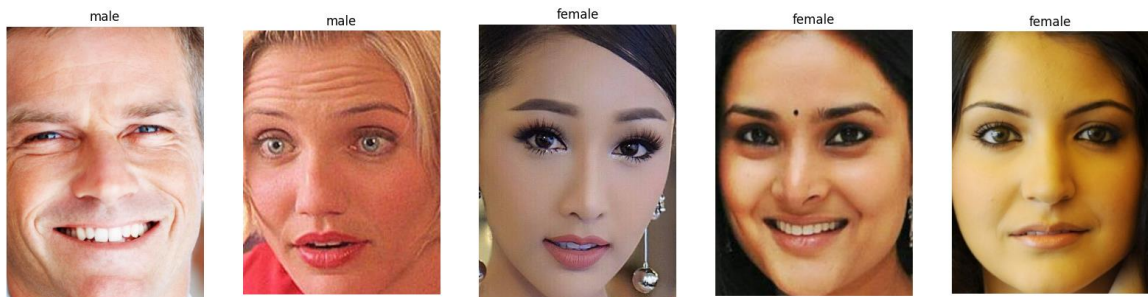
Hình 3-8 Kết quả test_4 trên tập test



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	5	0	100%
Test_2	4	1	80%
Test_3	3	2	60%
Test_4	5	0	100%
Test_5	5	0	100%
Test_6	5	0	100%
Test_7	4	1	80%
Test_8	5	0	100%
Test_9	5	0	100%
Test_10	4	1	80%

Trong bảng, qua các lần test, test_x có độ chính xác cao nhất là 100%, test_x có độ chính xác thấp nhất là 60%. Độ chính xác trung bình của 10 test là 90%.

3.7.2 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 2 Kết quả test_1 trên tập training



Hình 3-16 3 Kết quả test_2 trên tập training



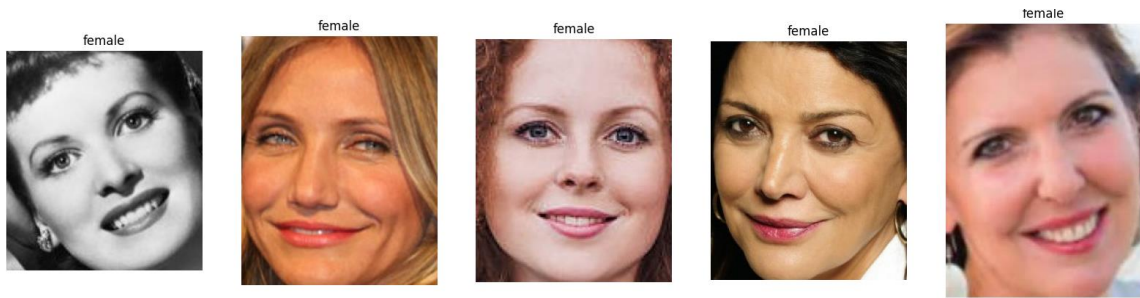
Hình 3-17 Kết quả test_3 trên tập training



Hình 3-18 Kết quả test_4 trên tập training



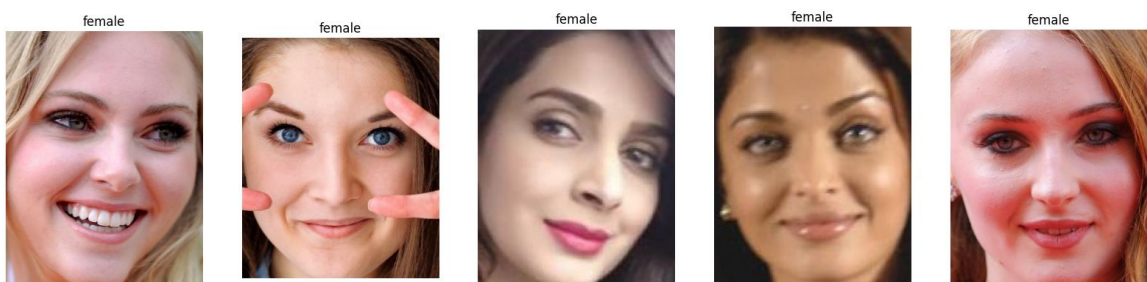
Hình 3-19 Kết quả test_5 trên tập training



Hình-19 Kết quả test_ 6 trên tập training



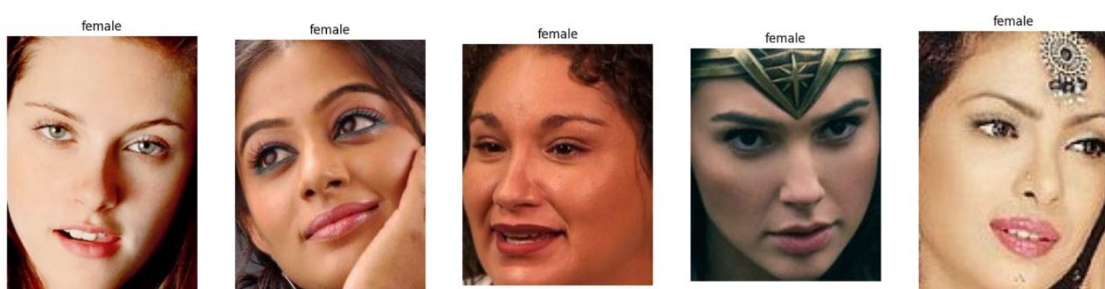
Hình 3-20 Kết quả test_ 7 trên tập training



Hình 3-21 Kết quả test_ 8 trên tập training



Hình – 3-22 Kết quả test_ 9 trên tập training



Hình 3-23 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

Với đề tài “Xây dựng mô hình học máy cho bài toán phân biệt giới tính dựa trên khuôn mặt” đồ án 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, ngôn ngữ lập trình Python, thư viện phụ trợ TensorFlow.
- Tìm hiểu về mạng CNN. Xây dựng mô hình CNN ứng dụng trong nhận diện khuôn mặt
- Thử nghiệm và đánh giá mô hình trong nhận diện khuôn mặt.

Một số hạn chế:

- 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

TÀI LIỆU THAM KHẢO

- [1.] <https://viblo.asia/p/tuan-1-gioi-thieu-xu-ly-anh-yMnKMdEQ57P>
- [2.] <https://vinbigdata.com/kham-pha/xu-ly-du-lieu-anh-mot-so-kien-thuc-can-ban.html>
- [3.] <https://viblo.asia/p/deep-learning-tim-hieu-ve-mang-tich-chap-cnn-maGK73bOKj2>
- [4.] <https://nttuan8.com/bai-5-gioi-thieu-ve-xu-ly-anh/>
- [5.] <https://www.kaggle.com/datasets/cashutosh/gender-classification-dataset>
- [6.] <https://www.elcom.com.vn/may-hoc-machine-learning-la-gi-ung-dung-cong-nghe-may-hoc-trong-thuc-tien-1666003970>