

ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Tìm hiểu và triển khai thử nghiệm phát hiện bất thường sử dụng hình ảnh

NGUYỄN ĐỨC QUÂN

Quan.nd193063@sis.hust.edu.vn

Ngành Điện tử viễn thông

Chuyên ngành Kỹ thuật máy tính

Giảng viên hướng dẫn: TS. Nguyễn Đình Văn

KHOA: Kỹ thuật truyền thông

HÀ NỘI, 10/2023

ĐỀ TÀI TỐT NGHIỆP

Tìm hiểu và triển khai thử nghiệm phát hiện bất thường sử dụng hình ảnh

Giáo viên hướng dẫn
Ký và ghi rõ họ tên

Lời cảm ơn

Trong thời gian làm đồ án tốt nghiệp, em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy Văn và gia đình.

Bởi vì đồ án lần này em thực hiện trong kỳ hè, thời gian cũng khá là ngắn, nó đã gây ra cho em nhiều áp lực, nhờ sự động viên của thầy Văn và cô vũ của gia đình đã giúp em vượt qua được những áp lực. Em xin gửi lời cảm ơn chân thành đến thầy Văn, người đã tận tình hướng dẫn, chỉ bảo em trong suốt quá trình làm đồ án.

Cuối cùng, em xin chân thành cảm ơn gia đình, đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên em trong suốt quá trình học tập và hoàn thành đồ án tốt nghiệp.

Tóm tắt nội dung đồ án

Trong đồ án này, sinh viên thực hiện nghiên cứu, thử nghiệm và áp dụng kỹ thuật học máy, học sâu vào xử lý hình ảnh, phát hiện bất thường trong các chi tiết kỹ thuật. Cụ thể, em đã tiến hành nghiên cứu tổng quan về các thuật toán học máy, các thuật toán học sâu. Bên cạnh đó, em đã tìm hiểu và đọc 06 bài báo khoa học tổng quan về bài toán phát hiện bất thường sử dụng hình ảnh. Từ đó, em cũng tìm hiểu 01 bài báo khoa học điển hình cho kết quả cao về phát hiện bất thường sử dụng hình ảnh. Cuối cùng, em đã thử nghiệm cài đặt thuật toán học sâu sử dụng cấu trúc autoencoder nhằm phát hiện hình ảnh bất thường trong bộ dữ liệu nhỏ của tập dữ liệu MVTec AD.

Sinh viên thực hiện

Ký và ghi rõ họ tên

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN.....	1
1.1 Mục tiêu	1
1.2 Phương pháp.....	1
1.3 Công việc thực hiện	2
CHƯƠNG 2. TÌM HIỂU LÝ THUYẾT.....	3
2.1 Tổng quan về học máy	3
2.1.1 Khái niệm về học máy	3
2.1.2 Các thành phần của học máy	3
2.1.3 Ứng dụng của Học máy	4
2.2 Tổng quan về học sâu.....	5
2.2.1 Khái niệm về học sâu	5
2.2.2 Các thành phần của học sâu	6
2.2.3 Ứng dụng học sâu	6
2.3 Transform trong học sâu	7
2.3.1 Các phương pháp biến đổi dữ liệu	7
2.3.2 Các phương pháp biến đổi không gian (Space Transformations)	8
2.4 Mạng nơ-ron tích chập (Convolutional Neural Networks).....	8
2.4.1 Lớp tích chập (Convolutional layer).....	8
2.4.2 Lớp tổng hợp (Pooling layer).....	11
2.4.3 Lớp kích hoạt (Activation layer).....	12
2.4.4 Lớp kết nối đầy đủ (Fully Connected Layer).....	13
2.4.5 Ưu điểm của CNNs	13
2.4.6 Ứng dụng của CNNs	13
2.5 Autoencoder	14
CHƯƠNG 3. TÌM HIỂU VỀ BÀI TOÁN PHÁT HIỆN BẤT THƯỜNG SỬ DỤNG HÌNH ẢNH	18
3.1 Nghiên cứu tổng quan	18
3.1.1 Bài báo số 1	18
3.1.2 Bài báo số 2	19
3.1.3 Bài báo số 3	20
3.1.4 Bài báo số 4	21
3.1.5 Bài báo số 5	22

3.1.6	Bài báo số 6.....	23
3.2	Nghiên cứu điển hình.....	23
CHƯƠNG 4. TRIỂN KHAI THUẬT TOÁN AUTOENCODER TRÊN TẬP DỮ LIỆU NHỎ CỦA MVC TEC		27
4.1	Bộ dữ liệu.....	27
4.2	Phương pháp sử dụng.....	27
4.3	Xây dựng thuật toán và kiểm thử.....	27
4.3.1	Mô hình autoencoder	28
4.4	Kiểm tra và đánh giá	28
CHƯƠNG 5. CHƯƠNG KẾT LUẬN		31
5.1	Đánh giá mức độ hoàn thiện	31
5.2	Hướng phát triển trong tương lai	32
TÀI LIỆU THAM KHẢO		33

DANH MỤC BẢNG BIỂU

Bảng 1. Bảng công việc dự kiến	2
Bảng 2 Bảng kết quả thực hiện	32

DANH MỤC HÌNH VẼ

Hình 2.1 Hình ảnh minh họa học máy	3
Hình 2.2 Ba cách học tập của học máy	4
Hình 2.3 Ứng dụng của học máy	5
Hình 2.4 Học sâu và học máy	6
Hình 2.5 Mô hình deep neural network	6
Hình 2.6 Ứng dụng của học sâu	7
Hình 2.7 Kernel kích thước 3x3	9
Hình 2.8 Ma trận A trong ma trận X	9
Hình 2.9 Tích chập giữa hai ma trận	9
Hình 2.10 Padding của một ma trận	10
Hình 2.11 Tổng hợp các kernel	10
Hình 2.12 Tích chập 3 chiều	11
Hình 2.13 Quá trình lớp tích chập	11
Hình 2.14 Pooling	12
Hình 2.15 Max pooling và average pooling	12
Hình 2.16 Hàm kích hoạt ReLU	13
Hình 2.17 Quá trình diễn ra trong CNNs	13
Hình 2.18 Tensor kích thước $a*b*h$	14
Hình 2.19 Chiều đi của autoencoder	15
Hình 2.20 Cấu trúc của autoencoder	15
Hình 2.21 Tổng quan quá trình autoencoder	16
Hình 3.1 Đối với mỗi danh mục tập dữ liệu, tỷ lệ các mẫu được phân loại chính xác Không có bất thường (hàng trên cùng) và hình ảnh bất thường (dưới cùng hàng) được đưa ra. Phương pháp có giá trị tb cao nhất được tô đậm cho mỗi hàng	20
Hình 3.2 So sánh các phương pháp phát hiện bất thường	21
Hình 3.3 Giá trị AUC khi thử nghiệm phương pháp MemAE	23
Hình 3.4 So sánh giá trị trung bình AUC với các phương pháp và các tập dữ liệu	24
Hình 3.5 AUC khi thay đổi siêu tham số	25
Hình 3.6 AUC khi thay đổi siêu tham số 2	25
Hình 3.7 Điểm bất thường ở lõi cáp	25
Hình 3.8 Điểm bất thường ở đai ốc	26
Hình 3.9 Điểm bất thường ở transistor	26
Hình 4.1 Lưu đồ thuật toán phát hiện bất thường bằng autoencoder	27
Hình 4.2 Mô hình autoencoder	28

Hình 4.3 Trước khi vào các lớp tích chập khác, hình ảnh được đưa qua các hàm giảm kích thước.....	28
Hình 4.4 Ví dụ về các thông số	29

CHƯƠNG 1. TỔNG QUAN

1.1 Mục tiêu

Phát hiện các lỗi, bất thường của các chi tiết công nghiệp sản xuất từ dây chuyền là một trong những vấn đề nan giải. Thông thường, các nhà máy cần có các công nhân, kỹ sư lành nghề, có khả năng phát hiện chính xác và nhanh chóng các chi tiết bất thường này. Tuy nhiên, do quy trình sản xuất tự động, quy mô sản xuất lớn, việc bỏ sót các lỗi bất thường vẫn tồn tại. Điều này gây ảnh hưởng đến chất lượng sản phẩm cũng như gây lãng phí tài nguyên. Đặc biệt, trong trường hợp của cá nhân em, được tham gia vào doanh nghiệp sản xuất thiết bị y tế, nhu cầu phát hiện bất thường càng trở nên cấp thiết. Sai sót trong việc phát hiện các bất thường này có thể gây hậu quả nghiêm trọng.

Nhằm nâng cao hiệu quả phát hiện bất thường trong các chi tiết công nghiệp, một trong những giải pháp được đề ra là sử dụng các camera chuyên dụng, chụp và phân tích dữ liệu hình ảnh của các chi tiết theo thời gian thực. Các mô hình học máy sẽ được ứng dụng nhằm phân loại và phát hiện bất thường trên các chi tiết này một cách tự động. Vì thế, trong khuôn khổ đề án này, em đặt ra mục tiêu tìm hiểu về bài toán phát hiện bất thường trong chi tiết công nghiệp sử dụng kỹ thuật học máy, học sâu.

Do điều kiện thời gian và kiến thức có hạn, em xin đặt ra các mục tiêu cho đề tài “Tìm hiểu và triển khai thử nghiệm phát hiện bất thường sử dụng hình ảnh” như sau:

- Tìm hiểu tổng quan về học máy, học sâu.
- Tìm hiểu tổng quan về bài toán phát hiện bất thường thông qua hình ảnh sử dụng học máy, học sâu
- Tìm hiểu một nghiên cứu điển hình về bài toán phát hiện bất thường thông qua hình ảnh sử dụng học sâu
- Triển khai, cài đặt thuật toán học sâu giúp phát hiện hình ảnh bất thường trên tập dữ liệu nhỏ của bộ dữ liệu MVTec.

1.2 Phương pháp

Để tìm hiểu tổng quan về học máy, học sâu, em đã đọc các tài liệu liên quan đến các thuật toán học máy như: các blog có nhiều bài viết về học máy, bài giảng của thầy Văn,.. Học sâu như: cộng đồng trực tuyến Viblo, các video chia sẻ kiến thức trên nền tảng Youtube. Từ đó đưa ra tổng kết và đánh giá của bản thân.

Để tìm hiểu tổng quan về bài toán “Tìm hiểu và triển khai thử nghiệm phát hiện bất thường sử dụng hình ảnh” em đã tìm và đọc các bài báo tổng quan về bài toán này giai đoạn từ năm 2019 đến năm 2022. Bên cạnh đó, em cũng tìm hiểu về bộ dữ liệu MVTec AD, bộ dữ liệu hình ảnh về các chi tiết có bất thường được sử dụng rộng rãi trong các nghiên cứu khoa học gần đây.

Từ các bài báo tổng quan, em đã chọn một nghiên cứu có kết quả cao trong phát hiện bất thường để tìm hiểu phương thức phát hiện và đánh giá kết quả một mô hình học máy khi áp dụng vào bài toán “Tìm hiểu và triển khai thử nghiệm phát hiện bất thường sử dụng hình ảnh”.

Cuối cùng, trên nền tảng pycharm, em đã thử nghiệm triển khai thuật toán phát hiện bất thường sử dụng cấu trúc autoencoder và tập dữ liệu MVTec.

1.3 Công việc thực hiện

STT	Tên công việc	Thời gian dự kiến	Nội dung
1	Tìm hiểu tổng quan học máy (ML), học sâu (DL),	01/08-08/08 (1 tuần)	Tìm hiểu về học máy, học sâu, mạng nơ-ron tích chập, autoencoder
2	Tìm hiểu các báo cáo tổng quan về bài toán phát hiện bất thường	09/08-16/08 (1 tuần)	Tìm và đọc một số bài báo tổng quan về phát hiện bất thường sử dụng hình ảnh.
3	Tìm hiểu nghiên cứu điển hình về bài toán phát hiện bất thường	17/08-01/09 (2 tuần)	Tìm hiểu bài báo “Cutpaste: Self-supervised learning for anomaly detection and localization”
4	Triển khai một bài toán phát hiện bất thường	02/09-23/09 (3 tuần)	Triển khai bài toán dựa trên MVTec anomaly dataset, xây dựng mạng neural nhân tạo CNN, sử dụng autoencoder
5	Viết báo cáo	24/09-01/10 (1 tuần)	Kiểm tra lại tổng quan đề tài và viết báo cáo

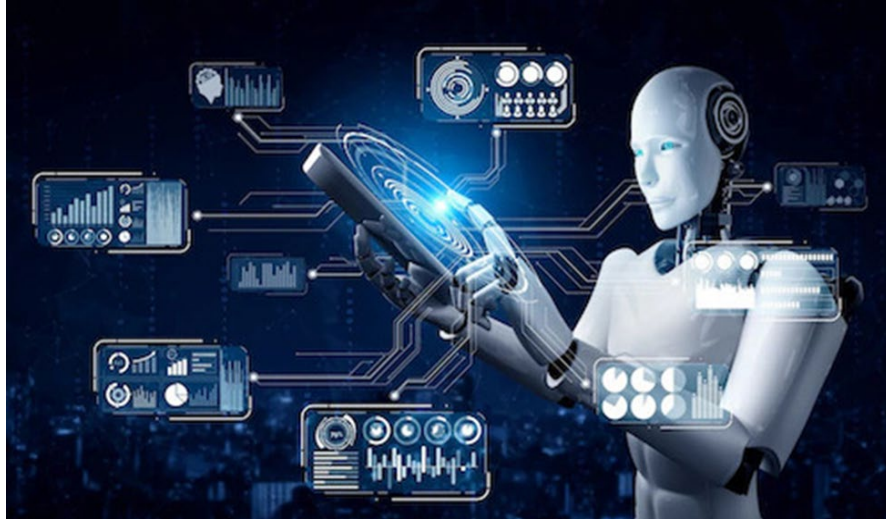
Bảng 1. Bảng công việc dự kiến

CHƯƠNG 2. TÌM HIỂU LÝ THUYẾT

2.1 Tổng quan về học máy

2.1.1 Khái niệm về học máy

Học máy là một lĩnh vực của trí tuệ nhân tạo (Artificial Intelligence) và khoa học máy tính, tập trung vào việc sử dụng dữ liệu và thuật toán để tự điều chỉnh, và từ đó nó có thể hoàn thành một nhiệm vụ cụ thể.



Hình 2.1 Hình ảnh minh họa học máy

Học máy dựa vào việc xử lý dữ liệu và phân tích dữ liệu để có thể tự động tối ưu hóa bản thân nó, từ đó làm tăng khả năng xử lý các nhiệm vụ mới hơn. Các bài toán phổ biến hiện nay của học máy là dự đoán hoặc phân loại dựa trên các dữ liệu được nó khai thác. Các bài toán dự đoán thường là giá nhà, giá xe, v.v, còn các bài toán phân loại thường là nhận diện chữ viết tay, đồ vật, v.v.

2.1.2 Các thành phần của học máy

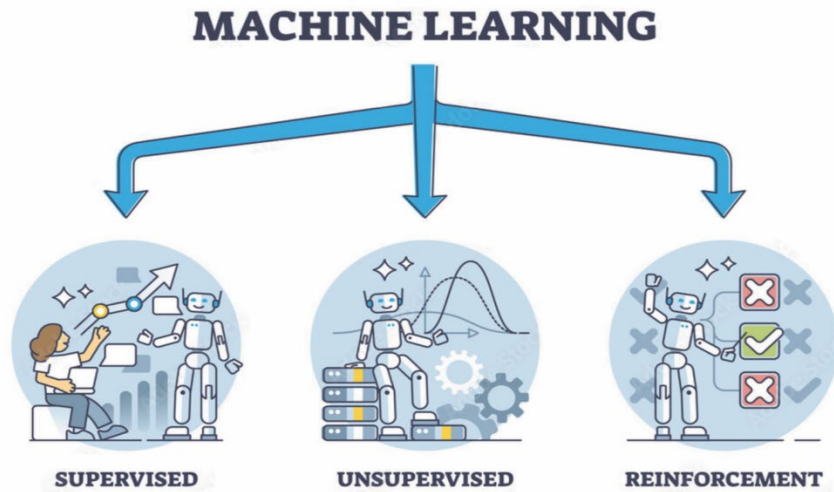
Học máy bao gồm các khái niệm và thành phần quan trọng sau:

Dữ liệu (Data): Dữ liệu hay còn gọi đầu vào cho mô hình là yếu tố cơ bản của học máy, nó cung cấp một lượng lớn thông tin hữu ích, có thể là văn bản, hình ảnh, âm thanh, hoặc bất cứ định dạng nào mà máy tính có thể hiểu.

Mô hình (Model): Đây là phần chính và quan trọng nhất của một học máy, mô hình tập hợp các cấu trúc neural, thuật toán để nhận đầu vào và xử lý nó để có thể thực hiện nhiệm vụ được yêu cầu như phân loại, dự đoán, phát hiện. Một học máy có mô hình càng tốt thì đầu ra kết quả của nhiệm vụ càng chính xác

Học tập (Learning): Là quá trình đưa vào mô hình một lượng lớn dữ liệu để mô hình có thể tìm ra mối quan hệ từ dữ liệu. Các loại học tập hiện nay là học có giám sát (Supervised learning), học không giám sát (Unsupervised learning) hoặc học tăng cường (Reinforcement learning), tùy thuộc vào loại nhiệm vụ và dữ liệu mà sử dụng cách học một cách hiệu quả nhất.

Ba cách học tập chính của Học máy:

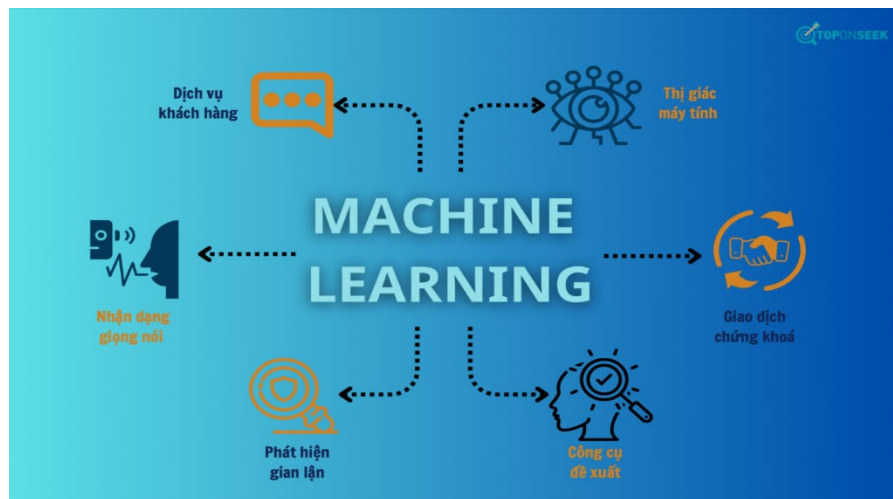


Hình 2.2 Ba cách học tập của học máy

- Học có giám sát: Đây là cách học mà dữ liệu đầu vào đã được gắn nhãn, nghĩa là với mỗi dữ liệu được đưa vào mô hình đều đã được phân loại một cách chính xác đầu ra mong muốn. Khi đó mô hình sẽ ánh xạ các cặp dữ liệu và nhãn tương ứng với nhau. Ví dụ như hình ảnh xe oto với nhãn là oto. Một số phương pháp được sử dụng trong học có giám sát bao gồm: logistic regression, neural networks, linear regression, naive bayes, random forest, và support vector machine (SVM)
- Học không giám sát: Đây là cách học mà dữ liệu đầu vào không được gắn nhãn, nghĩa là với mỗi dữ liệu được đưa vào mô hình đều phải cố gắng phân loại hoặc tìm ra các cấu trúc ẩn bên trong. Một số thuật toán thường được sử dụng trong học không giám sát như: k-means, HAC (Hierarchical Agglomerative Clustering), SOM (Self-Organizing Map), DBSCAN, FCM, ...
- Học tăng cường (Reinforcement Learning): Loại này thường được sử dụng trong việc đào tạo máy tính để thực hiện các tác vụ trong môi trường động. Mô hình tương tác với môi trường, nhận được phản hồi từ môi trường về hành động của nó, và điều chỉnh hành vi của mình để tối ưu hóa mục tiêu cụ thể.

2.1.3 Ứng dụng của Học máy

Học máy đã có sự ảnh hưởng lớn đến nhiều lĩnh vực và ngành công nghiệp, bao gồm:



Hình 2.3 Ứng dụng của học máy

Thị trường tài chính: Dự đoán giá cổ phiếu, phát hiện bất thường tài chính và tối ưu hóa quản lý rủi ro.

Y tế: Phát hiện bệnh, dự đoán kết quả điều trị, và phân tích hình ảnh y tế.

Công nghiệp sản xuất: Tối ưu hóa dây chuyền sản xuất, dự đoán thời gian bảo dưỡng máy móc, và kiểm tra chất lượng sản phẩm.

Ngôn ngữ tự nhiên: Xây dựng chatbot, dịch thuật và phân tích cảm xúc trong văn bản.

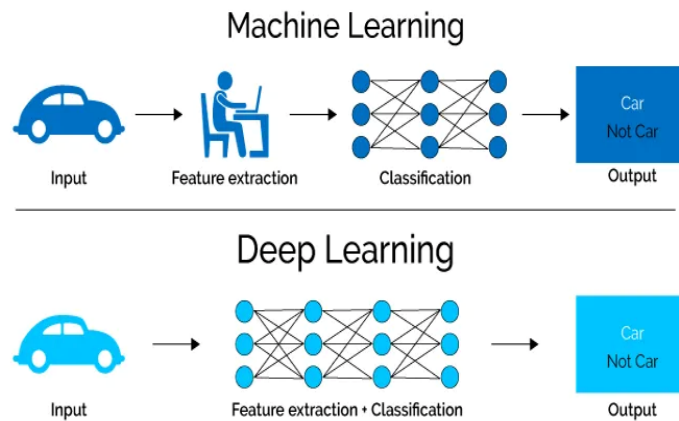
Tự động hóa và xe tự hành: Phát triển xe tự lái và robot tự động hoá.

Học máy đang phát triển nhanh chóng và có tiềm năng lớn trong tương lai. Tuy nhiên, cũng cần quan tâm đến vấn đề về đạo đức và quản lý dữ liệu trong quá trình sử dụng học máy. Sự phát triển của học máy sẽ tiếp tục làm thay đổi thế giới và tạo ra những cơ hội mới trong tương lai cho tất cả mọi người.

2.2 Tổng quan về học sâu

2.2.1 Khái niệm về học sâu

Deep Learning (DL), hay Học Sâu trong tiếng Việt, là một lĩnh vực quan trọng trong Học máy (ML) và trí tuệ nhân tạo (Artificial Intelligence - AI). DL tập trung vào việc sử dụng các mạng neural sâu (Deep neural networks) để học và biểu diễn dữ liệu phức tạp. DL cố gắng bắt chước khả năng tư duy và suy nghĩ của một bộ não như con người. Mạng neural trong DL gồm nhiều lớp khác nhau, càng có nhiều lớp thì mạng càng “sâu”. Học sâu thúc đẩy nhiều ứng dụng và dịch vụ trí tuệ nhân tạo (AI) nhằm cải thiện tự động hóa, thực hiện các tác vụ phân tích và vật lý mà không cần sự can thiệp của con người.



Hình 2.4 Học sâu và học máy

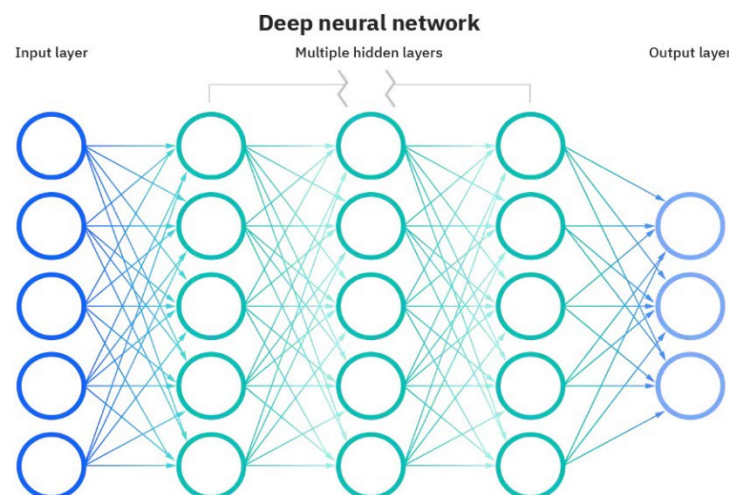
2.2.2 Các thành phần của học sâu

Học sâu bao gồm các yếu tố quan trọng sau:

Mạng Neural Sâu (Deep Neural Networks): Điểm nổi bật của Học sâu là sử dụng các mạng neural sâu, tập hợp các node (các neural nhân tạo) được tổ chức thành các lớp (layers), có nhiều lớp neural chồng lên nhau. Các node nhận dữ liệu đầu vào và truyền dữ liệu đầu ra cho các node tiếp theo.

Học Tự Động (Feature Learning): Khác với Học máy thì Học sâu có khả năng tự động học các đặc trưng quan trọng từ dữ liệu đầu vào. Điều này giúp giảm thiểu công việc của con người khi trích xuất các đặc trưng từ đầu vào. Quá trình này gắn liền với khả năng tự tối ưu hóa của mô hình.

Dữ liệu lớn (Big Data): Để có thể đạt được kết quả tốt nhất thì Học sâu yêu cầu lượng dữ liệu đầu vào là cực lớn, càng có nhiều dữ liệu thì mô hình Học sâu càng có khả năng tự học và tự điều chỉnh tốt hơn. Đây cũng là vấn đề lớn khi thu thập dữ liệu cho mô hình.



Hình 2.5 Mô hình deep neural network

2.2.3 Ứng dụng học sâu

Nhận diện hình ảnh: Học sâu đã giúp nâng cao hiệu suất trong việc nhận diện đối tượng và khuôn mặt trong hình ảnh.

Xử lý ngôn ngữ tự nhiên (NLP): Học sâu đã cải thiện hiệu suất trong các nhiệm vụ như dự đoán từ vựng, dịch máy, và phân tích cảm xúc từ văn bản.

Tự động hóa và robot tự động hoá: Học sâu được sử dụng trong xe tự lái, robot tự động hoá và quản lý dây chuyền sản xuất.

Y tế: Học sâu được sử dụng trong việc dự đoán bệnh, phát hiện tế bào ác tính, và phân tích hình ảnh y tế.



Hình 2.6 Ứng dụng của học sâu

Học sâu vẫn đang phát triển và có tiềm năng lớn. Học sâu đòi hỏi cơ sở hạ tầng máy tính mạnh mẽ để đào tạo và triển khai các mô hình lớn. Các nghiên cứu mới về mạng neural sâu và học tăng cường đang mở ra nhiều cơ hội mới. Tuy nhiên, cũng cần quan tâm đến vấn đề về đạo đức và đảm bảo tính riêng tư trong việc sử dụng DL.

2.3 Transform trong học sâu

Transform trong Học sâu đóng vai trò quan trọng trong việc biến đổi dữ liệu đầu vào hoặc biểu diễn của dữ liệu để cải thiện khả năng học của mô hình. Với mỗi dữ liệu đầu vào khi đi qua transform thì sẽ cho ra một dữ liệu mới, để mô hình có thể học được thêm những dữ liệu mới. Đây cũng là một giải pháp cực kỳ hữu ích với những nguồn dữ liệu có hạn, nó tạo ra một số biến thể của cái gốc.

2.3.1 Các phương pháp biến đổi dữ liệu

- Chuẩn hóa (Normalization)

Chuẩn hóa là một biến đổi cơ bản trong Học sâu. Nó bao gồm việc đưa dữ liệu về miền giá trị cụ thể, thường là giữa 0 và 1 hoặc -1 và 1. Điều này giúp cân bằng tỷ lệ giữa các đặc trưng và giúp mô hình học nhanh hơn. Ví dụ, trong mạng neural, chuẩn hóa có thể thực hiện thông qua việc trừ đi giá trị trung bình và chia cho độ lệch chuẩn của dữ liệu.

- Data Augmentation

Data Augmentation là một phương pháp mạnh mẽ và là một trong những cách phổ biến được áp dụng “transform” trong xử lý hình ảnh. Nó bao gồm tạo ra các biến thể của dữ liệu bằng cách thực hiện các phép biến đổi như xoay, thu phóng, cắt, hoặc thay đổi ánh sáng một cách ngẫu nhiên hoặc theo một số quy tắc. Việc

này không chỉ tạo ra thêm dữ liệu huấn luyện, mà còn giúp mô hình học được các biến thể của dữ liệu và tránh việc overfitting.

Biến đổi phi tuyến tính (Non-linear Transformations)

Biến đổi phi tuyến tính thường được sử dụng để biến đổi đặc trưng của dữ liệu thành các biểu diễn phức tạp hơn. Ví dụ, việc áp dụng hàm kích hoạt phi tuyến tính như hàm ReLU (Rectified Linear Unit) có thể giúp mô hình học được biểu diễn phi tuyến tính của dữ liệu.

Domain Transformation (Biến Đổi Miền): Trong việc xử lý dữ liệu từ các miền khác nhau, có thể áp dụng các biến đổi để chuyển đổi dữ liệu từ một miền này sang miền khác. Ví dụ, trong xử lý hình ảnh y khoa, chuyển đổi hình ảnh một chiều (2D) sang ảnh ba chiều (3D) để sử dụng cho việc phân loại bệnh lý.

2.3.2 Các phương pháp biến đổi không gian (Space Transformations)

- Tích chập (Convolution)

Tích chập là một biến đổi quan trọng trong xử lý hình ảnh và xử lý chuỗi. Nó được sử dụng để trích xuất các đặc trưng từ dữ liệu bằng cách áp dụng một bộ lọc (kernel) trượt qua dữ liệu. Việc này giúp mô hình nhận biết các đặc điểm cục bộ trong dữ liệu, chẳng hạn như cạnh, góc, hoặc mẫu chuỗi.

- Tích hợp (Pooling)

Tích hợp là một phép biến đổi không gian thường được sử dụng sau Convolution. Nó giúp giảm kích thước của biểu diễn không gian, giúp giảm chi phí tính toán và tránh overfitting. Phép tích hợp thông thường bao gồm việc lấy giá trị lớn nhất (Max Pooling) hoặc lấy giá trị trung bình (Average Pooling) trong một vùng cụ thể của dữ liệu.

Tất cả các loại biến đổi trong transform có thể giúp cải thiện hiệu suất của mô hình Học sâu và làm cho nó phù hợp hơn với mục tiêu cụ thể của mỗi bài toán. Sự lựa chọn của biến đổi phụ thuộc vào bài toán cụ thể và loại dữ liệu đang làm việc.

2.4 Mạng nơ-ron tích chập (Convolutional Neural Networks)

Mạng nơ-ron tích chập (CNNs), là kiến trúc mạng neural nhân tạo nâng cao, được xây dựng để giải quyết các bài toán phức tạp, dữ liệu có cấu trúc rộng, đặc biệt là các bài toán liên quan đến xử lý hình ảnh, chẳng hạn như nhận diện đối tượng, phân loại hình ảnh, thậm chí là tự động lái xe.

Cấu trúc cơ bản của CNNs gồm các lớp.

2.4.1 Lớp tích chập (Convolutional layer)

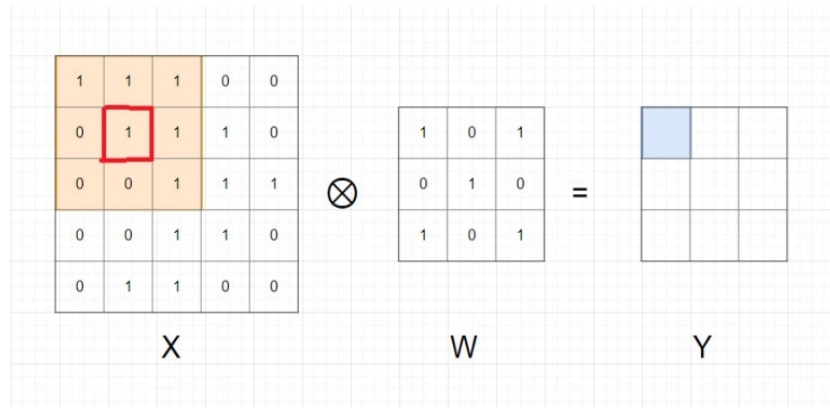
Lớp này sử dụng các bộ lọc (kernels) để thực hiện các phép tích chập dựa trên ảnh đầu vào để cho ra tín hiệu mới giảm bớt các đặc trưng mà không cần thiết, giữ lại các đặc trưng chính và quan trọng. Kết quả của 2 phép tích chập ma trận là một ma trận mới chứa các đặc trưng của ma trận đầu vào như ví dụ bên dưới:

Ta có một ma trận đầu vào là X , ta định nghĩa kernel là một ma trận vuông kích thước $k \times k$ trong đó k là số lẻ. k có thể bằng 1, 3, 5, 7, 9, ... Ví dụ kernel kích thước 3×3 như sau:

$$W = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

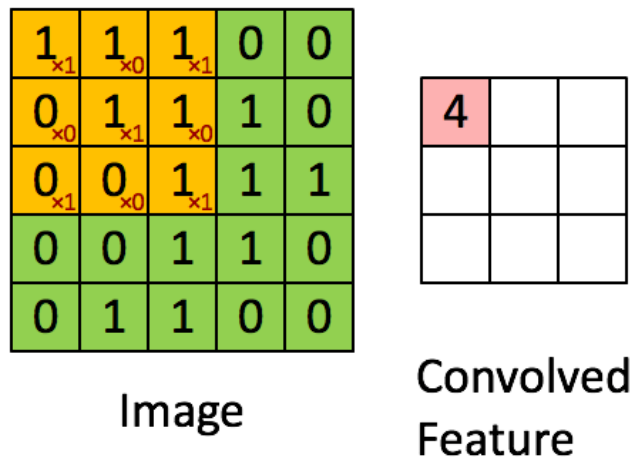
Hình 2.7 Kernel kích thước 3x3

Với mỗi phần tử xij trong ma trận X lấy ra một ma trận có kích thước bằng kích thước của kernel W (lấy phần tử xij làm trung tâm) gọi là ma trận A. Sau đó tính tổng các phần tử của phép tính tích chập của ma trận A với ma trận W, rồi viết vào ma trận kết quả Y như Hình 2.8.



Hình 2.8 Ma trận A trong ma trận X

Để tính giá trị của x22 ta nhân từng giá trị của ma trận A (được tô màu cam) với ma trận W, kết quả được ma trận Y. Tương tự với các giá trị khác trong ma trận X để ra được kết quả sau cùng. Để dễ hình dung ta có ví dụ bên dưới:









Hình 2.9 Tích chập giữa hai ma trận

Ta bắt gặp một vấn đề mới là các giá trị viền ngoài như x11, x12, x13, ... thì không thể tính được bởi không có ma trận A thỏa mãn bao bọc được các giá trị viền làm trung tâm, vậy nên ta thêm một lớp bọc bên ngoài gọi là padding để giải quyết vấn đề trên như Hình 2.10.

0	0	0	0	0	0	0
0	1	1	1	0	0	0
0	0	1	1	1	0	0
0	0	0	1	1	1	0
0	0	0	1	1	0	0
0	0	1	1	0	0	0
0	0	0	0	0	0	0

Hình 2.10 Padding của một ma trận

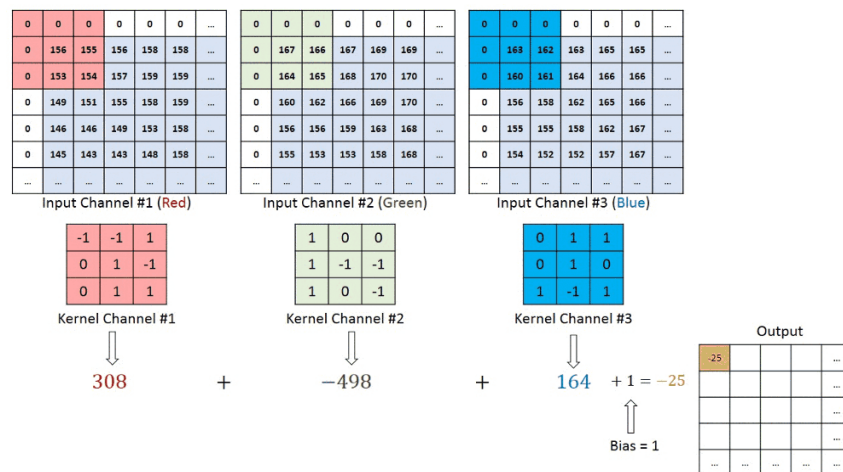
Mục đích của phép tính convolution trên ảnh là làm mờ, làm nét ảnh, xác định các đường, ... Mỗi kernel khác nhau thì phép tính tích chập sẽ có ý nghĩa khác nhau. Vì vậy tùy từng bài toán hoặc từng mục đích mà chúng ta chọn kernel cho phù hợp:

Operation	Kernel ω	Image result $g(x,y)$
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

Hình 2.11 Tổng hợp các kernel

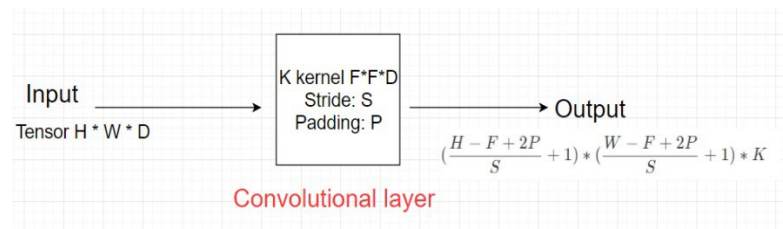
Đối với các ảnh màu có tới 3 kênh là red, green, blue (R, G, B) thì ta phải thực hiện phép tích chập bằng cách nhân một tensor 3 chiều với một kernel 3 chiều, sau khi thực hiện phép tích chập 3 chiều ta cộng thêm một giá trị gọi là “bias”.

Bias là một giá trị số được thêm vào sau khi tích chập để tạo ra tính phi tuyến tính trong mạng neural. Nó cho phép mô hình học và biểu diễn các đặc trưng tại các vị trí không phụ thuộc vào dữ liệu đầu vào. Trong trường hợp mạng tích chập, mỗi kênh đầu ra của tích chập sẽ có một bias riêng. Chúng ta chỉ cần khởi tạo bias và giá trị bias sẽ được thực hiện tự động bởi các thư viện và framework như TensorFlow, PyTorch, Keras, Cách thực hiện phép tích chập 3 chiều được thể hiện như Hình 2.12.



Hình 2.12 Tích chập 3 chiều

Tổng quát lại quá trình convolution layer:



Hình 2.13 Quá trình lớp tích chập

Đầu ra của tích chập lớp sẽ qua hàm activation function trước khi trở thành đầu vào của lớp tích chập tiếp theo.

Tổng số parameter của layer: Mỗi kernel có kích thước $F \times F \times D$ và có 1 hệ số bias, nên tổng parameter của 1 kernel là $F \times F \times D + 1$. Mà lớp tích chập áp dụng K kernel \Rightarrow Tổng số parameter trong layer là $K \times (F \times F \times D + 1)$.

2.4.2 Lớp tổng hợp (Pooling layer)

Lớp tổng hợp giúp giảm kích thước khối ma trận đầu vào thông qua việc tìm ra 1 giá trị đại diện cho mỗi vùng không gian mà bộ lọc đi qua, không làm thay đổi đường nét chính của bức ảnh nhưng giảm được kích thước. Lớp tổng hợp thường được thực hiện sau các lớp tích chập. Lớp này giúp giảm độ phức tạp của mô hình và giúp tránh overfitting.

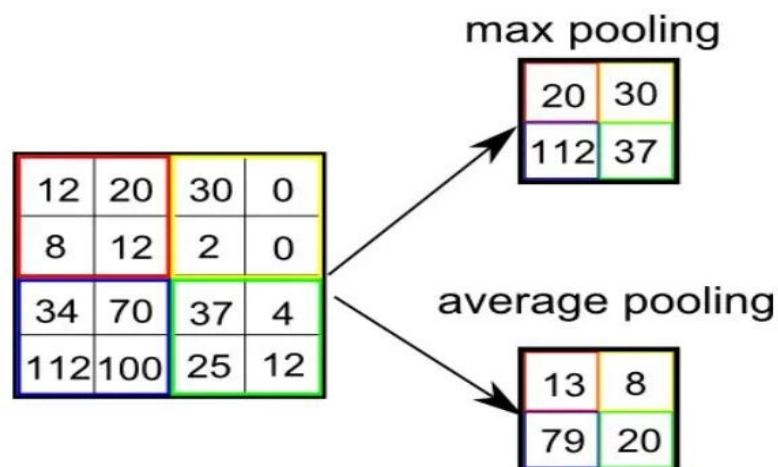
Ta có pooling size kích thước $K \times K$. Đầu vào của lớp pooling là ma trận có kích thước $H \times W \times D$, tách ma trận này ra làm D ma trận kích thước $H \times W$. Với mỗi ma trận $H \times W$, vùng kích thước $K \times K$ trên ma trận $H \times W$ ta tìm maximum hoặc average của dữ liệu rồi viết vào ma trận kết quả.

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

Hình 2.14 Pooling

Có 2 loại pooling layer phổ biến là: max pooling và average pooling. Max pooling thì trong ma trận kích thước $K \times K$ ta chọn phần tử có giá trị lớn nhất. Average pooling thì ta tính trung bình cộng của các phần tử trong ma trận kích thước $K \times K$.



Hình 2.15 Max pooling và average pooling

2.4.3 Lớp kích hoạt (Activation layer)

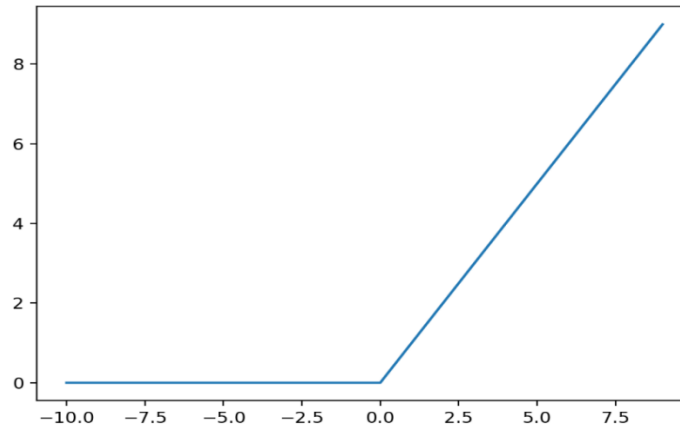
Các lớp này thường được sử dụng hàm kích hoạt như ReLU (Rectified Linear Unit) để thêm tính phi tuyến tính vào mô hình

Hàm ReLU đặc trưng bởi cách xử lý dữ liệu đầu vào và tạo ra trị đầu ra trên 1 quy tắc đơn giản:

$$\text{ReLU}(x) = \max(0, x)$$

Trong đó:

x là giá trị đầu vào, và giá trị đầu ra sẽ là 0 nếu x âm, hoặc trả về x nếu x lớn hơn hoặc bằng 0



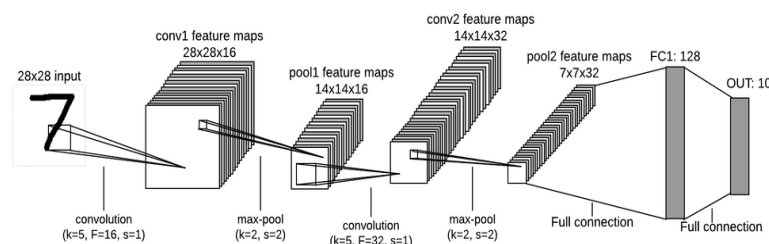
Hình 2.16 Hàm kích hoạt ReLU

Hàm ReLU tính toán nhanh chóng vì nó chỉ thực hiện một phép so sánh và 1 phép gán, không đòi hỏi nhiều như các hàm kích hoạt Sigmoid hay Tanh.

2.4.4 Lớp kết nối đầy đủ (Fully Connected Layer)

Sau khi qua các lớp tích chập và tổng hợp, các đặc trưng được duyệt qua các lớp kết nối đầy đủ để thực hiện phân loại hoặc dự đoán.

Một quá trình diễn ra trong CNNs được mô tả như Hình 2.17:



Hình 2.17 Quá trình diễn ra trong CNNs

2.4.5 Ưu điểm của CNNs

- Khả năng học đặc trưng tự động: CNNs có khả năng tự động học và trích xuất các đặc trưng quan trọng từ dữ liệu, giúp giảm bớt sự phụ thuộc vào việc tạo ra các đặc trưng thủ công bởi con người.
- Invariance về dịch chuyển: CNNs có khả năng nhận diện đối tượng trong ảnh bất kể vị trí của chúng, điều này có ích trong nhiều ứng dụng thực tế.
- Hiệu suất cao trên dữ liệu hình ảnh: CNNs đã đạt được hiệu suất xuất sắc trong nhiều cuộc thi và ứng dụng thực tế, chẳng hạn như phát hiện khuôn mặt, phân loại hình ảnh và xe tự động lái.

2.4.6 Ứng dụng của CNNs

- Nhận diện đối tượng: CNNs có thể được sử dụng để nhận diện đối tượng trong ảnh và video, từ nhận dạng khuôn mặt đến phát hiện xe hơi.
- Phân loại hình ảnh: CNNs có thể phân loại ảnh vào các lớp khác nhau, ví dụ như phân loại loại động vật hoặc thực phẩm.

- Xử lý ảnh y tế: Trong lĩnh vực y tế, CNNs đã được sử dụng để xác định bệnh từ hình ảnh chẩn đoán như tia X hoặc hình ảnh MRI.
- Xe tự động lái: CNNs đóng một vai trò quan trọng trong phát triển công nghệ xe tự động lái, giúp xe ô tô tự động nhận biết và phản ứng với môi trường xung quanh.

Một số hãng nổi tiếng đang sử dụng CNNs như: Apple đã sử dụng CNNs trong Face ID, công nghệ nhận dạng khuôn mặt trên các phiên bản Iphone mới nhất. Google sử dụng CNNs vào Google Photos để phân loại và tìm kiếm hình ảnh dựa trên nội dung, và cũng đang được áp dụng vào Waymo, công ty con của Alphabet để phát triển công nghệ xe tự động lái. Facebook cũng sử dụng CNNs để nhận dạng khuôn mặt và phân loại hình ảnh trên nền tảng của họ. NVIDIA công ty chuyên về đồ họa cũng sử dụng CNNs cho mục đích xử lý hình ảnh và thời gian thực bao gồm các giải pháp tự động lái xe.

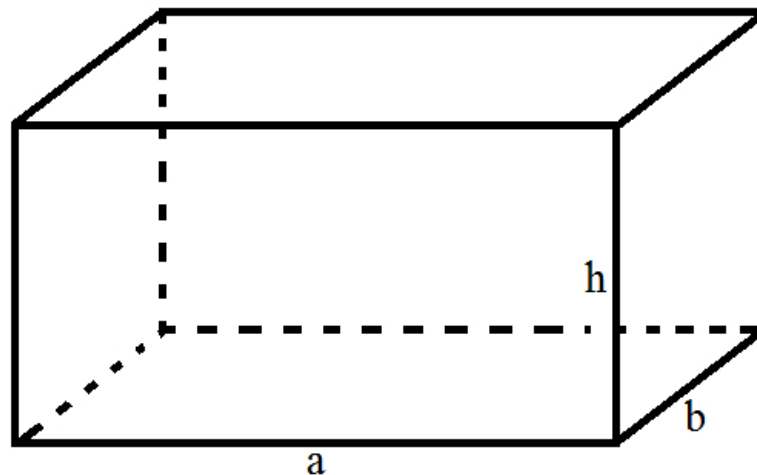
2.5 Autoencoder

Khi dữ liệu biểu diễn dạng 1 chiều, người ta gọi là vector, mặc định khi viết vector sẽ viết dưới dạng cột.

Khi dữ liệu dạng 2 chiều, người ta gọi là ma trận, kích thước là số hàng * số cột.

Khi dữ liệu nhiều hơn 2 chiều thì sẽ được gọi là tensor, ví dụ như dữ liệu có 3 chiều.

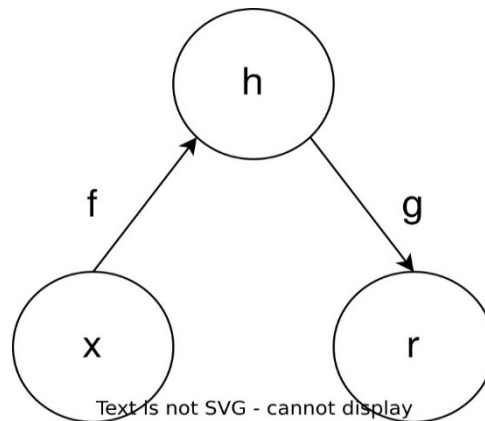
Để ý thì thấy là ma trận là sự kết hợp của các vector cùng kích thước. Xếp n vector kích thước m cạnh nhau thì sẽ được ma trận $m*n$. Thì tensor 3 chiều cũng là sự kết hợp của các ma trận cùng kích thước, xếp k ma trận kích thước $m*n$ lên nhau sẽ được tensor kích thước $m*n*k$.



Hình 2.18 Tensor kích thước $a*b*h$

Autoencoder là một kiến trúc mạng thần kinh nhân tạo trong học sâu được sử dụng trong nhiều nhiệm vụ, đặc biệt trong việc trích xuất đặc trưng và giảm chiều dữ liệu. Mục đích của autoencoder là tạo ra giá trị output gần đúng bằng cách chỉ tập trung vào các đặc trưng cần thiết. Trong thực tế, một autoencoder tập hợp các hạn chế buộc mạng neural phải tìm hiểu những cách thức mới để tái tạo cho dữ liệu, khác với chỉ đơn thuần là sao chép đầu ra.

Một autoencoder điển hình được xác định với một đầu vào, một đại diện bên trong và một đầu ra (đầu ra này xấp xỉ với đầu vào). Việc học tập xảy ra trong các layers gắn với biểu diễn bên trong.



Hình 2.19 Chiều đi của autoencoder

Một autoencoder cơ bản bao gồm hai phần chính:

Encoder (Bộ mã hóa): Phần này ánh xạ dữ liệu đầu vào vào không gian ẩn (latent space). Nó thường bao gồm một hoặc nhiều lớp neural để giảm chiều dữ liệu và biểu diễn nó trong không gian ẩn. Quá trình này tương tự như việc nén dữ liệu.

$$S = E(x)$$

Trong đó x là input Data, E là encoder, và s là output trong không gian ẩn

Decoder (Bộ giải mã): Phần này ánh xạ từ không gian ẩn trở lại không gian đầu vào. Nó có nhiệm vụ tái tạo dữ liệu từ biểu diễn ẩn làm sao càng giống càng tốt. Decoder thường chứa các lớp neural để phục hồi dữ liệu ban đầu từ không gian ẩn.

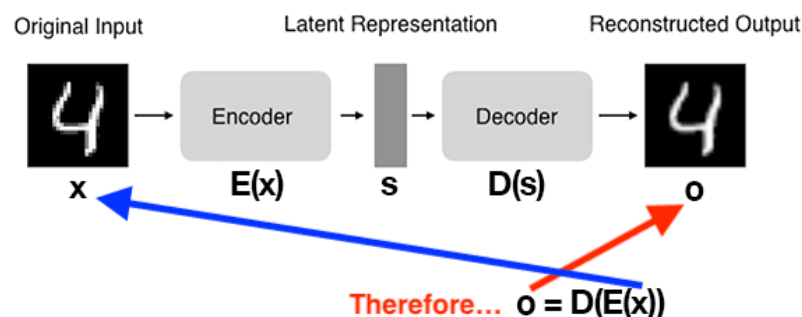
$$O = D(s)$$

Trong đó o là output cho Decoder D

Công thức chung cho toàn bộ quá trình Autoencoder là:

$$O = D(E(x))$$

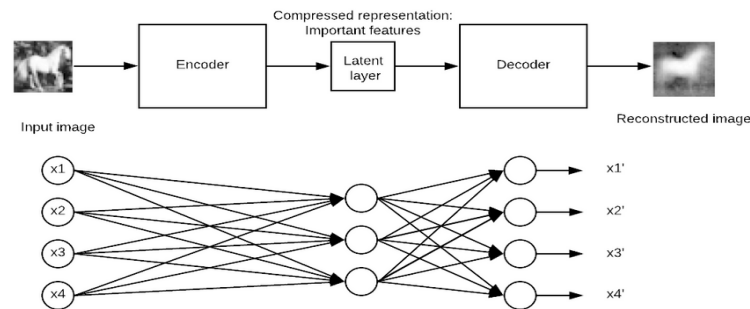
Được minh họa như sau:



Hình 2.20 Cấu trúc của autoencoder

Trên thực tế, có hai khối layers chính trông giống như một mạng neural truyền thống. Sự khác biệt nhỏ là layer chứa đầu ra phải bằng với đầu vào. Trong hình bên dưới, đầu vào ban đầu đi vào khối đầu tiên được gọi là encoder. Quá trình

encoder diễn ra trích xuất đặc trưng và giảm chiều dữ liệu. Biểu diễn bên trong này nén (giảm) kích thước của đầu vào. Trong khối thứ hai xảy ra việc tái tạo đầu vào. Giai đoạn cố gắng tái tạo lại dựa vào những đặc trưng được cung cấp ở khối lưu trữ. Đây là giai đoạn giải mã:



Hình 2.21 Tổng quan quá trình autoencoder

Autoencoder cố gắng học biểu diễn sao cho đầu ra của quá trình giải mã gần giống với dữ liệu đầu vào ban đầu. Điều này thúc đẩy mô hình học cách biểu diễn dữ liệu một cách hiệu quả bằng cách tìm ra các đặc trưng quan trọng và loại bỏ nhiễu.

Ứng dụng của autoencoders bao gồm:

- Nén dữ liệu: Autoencoders có thể được sử dụng để nén dữ liệu mà vẫn giữ lại thông tin quan trọng. Ví dụ, trong nén hình ảnh để giảm dung lượng lưu trữ.
- Phát hiện nhiễu và ngoại lai: Autoencoders có khả năng phát hiện nhiễu hoặc các điểm dữ liệu ngoại lai bằng cách so sánh đầu ra của mô hình với dữ liệu ban đầu.
- Trích xuất đặc trưng: Autoencoders có thể được sử dụng để trích xuất các đặc trưng quan trọng từ dữ liệu, chẳng hạn như trong việc phân loại hình ảnh hoặc âm thanh.

Hyperparameter là các thông số được định nghĩa trước khi huấn luyện mô hình, bao gồm:

Learning rate: là 1 siêu tham số quan trọng, giúp 1 mô hình hội tụ đúng cách, tránh các vấn đề như học quá nhanh hoặc quá chậm

(thuyết trình: LR là 1 số dương, giá trị nhỏ, thường là 0,01 hoặc là 0,001, LR quyết định khoảng cách mà các trọng số sẽ được cập nhật trong mỗi bước huấn luyện. hội tụ là khi mô hình học đủ tốt để có thể áp dụng được vào các dữ liệu mới)

Epoch: là số lần lặp

Batch_size: số lượng mẫu dữ liệu được sử dụng trong mỗi lần cập nhật trọng số

Kiến trúc: kiến trúc của mô hình gồm số lớp, số lượng đơn vị ẩn, hàm kích hoạt, ...

Regularization: các tham số tránh overfitting

Optimizer: các hàm tối ưu như Adam, RMSprop, SGD, ...

Loss function: MSE, ...

Autoencoders là một phần quan trọng của Học sâu và có nhiều biến thể như Variational Autoencoders (VAEs) và Denoising Autoencoders (DAEs), đã đóng góp vào nhiều thành công trong việc trích xuất đặc trưng và xử lý dữ liệu phức tạp, mở ra nhiều cơ hội cho nghiên cứu và ứng dụng trong nhiều lĩnh vực khác nhau.

CHƯƠNG 3. TÌM HIỂU VỀ BÀI TOÁN PHÁT HIỆN BẤT THƯỜNG SỬ DỤNG HÌNH ẢNH

3.1 Nghiên cứu tổng quan

Xử lý hình ảnh là một lĩnh vực quan trọng trong khoa học máy tính và trí tuệ nhân tạo, chuyên về việc xử lý, phân tích và trích xuất thông tin từ hình ảnh hoặc video. Đây là một phần quan trọng của thị giác máy tính, nơi máy tính được sử dụng để nhận biết và hiểu hình ảnh theo cách mà con người có thể.

Xử lý hình ảnh được sử dụng để: nhận diện đối tượng, xử lý ảnh y tế, thị giác máy tính, tự động hóa công việc, xử lý ảnh nghệ thuật

Xử lý hình ảnh là một lĩnh vực đa dạng và phát triển nhanh với nhiều ứng dụng thú vị trong cuộc sống hàng ngày và công nghiệp. Nó đã đóng một vai trò quan trọng trong việc cải thiện hiệu suất và hiệu quả trong nhiều ngành khác nhau.

Một số bài báo có phân tích về việc sử dụng xử lý hình ảnh để nhận biết bất thường.

3.1.1 Bài báo số 1

Bài báo "A Survey of Visual Sensory Anomaly Detection" [1] là một cuộc khảo sát tổng quan về lĩnh vực phát hiện sự bất thường sử dụng hình ảnh và video.

Bài báo đề xuất phân loại các bài toán phát hiện bất thường theo ba cấp độ:

Object-level AD (phát hiện bất thường cấp độ đối tượng): là bất thường xuất hiện trong một phần nhỏ của đối tượng

Scene-level AD (cấp độ cảnh): là bất thường xuất hiện trong một hình ảnh, trong đó bất thường là một đối tượng, ví dụ như là chướng ngại vật xuất hiện ở trên đường

Event-level AD (cấp độ sự kiện): là các sự kiện bất thường xuất hiện trong video. Dựa vào thông tin về các frames ảnh liên tục, hệ thống có thể phát hiện các chuyển động bất thường.

Và được chia ra thành 3 phương pháp học máy phát hiện bất thường là unsupervised (không giám sát), semi-supervised (bán giám sát) và supervised (giám sát)

Như đã giải thích từ trước supervised có nghĩa là bộ dữ liệu đã có nhãn và unsupervised là học không có nhãn, còn “semi-supervised learning”: là sử dụng dữ liệu được gán 1 phần nhãn hoặc nhãn không chính xác để huấn luyện mô hình. Ví dụ trong tập dữ liệu hình ảnh có nhiều con vật, bạn gán nhãn cho dữ liệu là chó nhưng vị trí chính xác của chó nằm trong ảnh thì không biết, đây gọi là nhãn yếu. Hoặc có thể là tập dữ liệu được dán nhãn nhưng lại không biết cái nhãn nào đúng, cái nhãn nào sai.

Bài báo đã đưa ra những thách thức mà lĩnh vực thị giác đang gặp phải và hướng phát triển trong tương lai. Các thách thức bao gồm khả năng xử lý sự bất thường trong các tình huống phức tạp và cải thiện hiệu suất phát hiện sự bất thường trong thời gian thực. Hướng phát triển tương lai gồm việc nghiên cứu và phát triển các

phương pháp học không giám sát, học yếu giám sát và học hoàn toàn giám sát hiệu quả hơn.

3.1.2 Bài báo số 2

Để kiểm thử một mô hình thì cần một lượng lớn dữ liệu bình thường và bất thường, nhưng việc này hầu như rất khó đối với những người còn mới để có được tập dữ liệu như vậy.

Bài báo "MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection" [2] giới thiệu một tập dữ liệu quan trọng được gọi là "MVTec AD". Nó chứa 5354 hình ảnh có độ phân giải cao được chia thành 15 loại đối tượng và kết cấu khác nhau, bao gồm 3629 hình ảnh không khiếm khuyết để đào tạo mô hình và 1725 hình ảnh để thử nghiệm. Tập dữ liệu này chứa một lượng lớn các danh mục như chai thủy tinh, đèn pha, đinh vít, nắp, viên thuốc, thảm, lưới, da thú, hạt phỉ, đai ốc kim loại, ngói, bàn chải đánh răng, transistor, gỗ và dây kéo. Mỗi danh mục bao gồm một tập hợp các hình ảnh huấn luyện không có khiếm khuyết và một tập hình ảnh thử nghiệm có nhiều loại khiếm khuyết khác nhau cũng như các hình ảnh không có khiếm khuyết. Tập dữ liệu được thiết kế đặc biệt cho mục tiêu phát hiện sự bất thường (anomaly detection) trong các tình huống thực tế.

Với việc sử dụng Convolutional autoencoder (CAEs), Convolutional neural network để phát hiện dị thường với bộ dữ liệu của MVTec đã cho ra kết quả với tỷ lệ các mẫu phát hiện bất thường là không tốt bằng so với việc phát hiện không bất thường.

	Category	AE (SSIM)	AE (L2)	AnoGAN	CNN Feature Dictionary	Texture Inspection	Variation Model
Textures	Carpet	0.43	0.57	0.82	0.89	0.57	-
		0.90	0.42	0.16	0.36	0.61	-
	Grid	0.38	0.57	0.90	0.57	1.00	-
		1.00	0.98	0.12	0.33	0.05	-
	Leather	0.00	0.06	0.91	0.63	0.00	-
		0.92	0.82	0.12	0.71	0.99	-
Objects	Tile	1.00	1.00	0.97	0.97	1.00	-
		0.04	0.54	0.05	0.44	0.43	-
	Wood	0.84	1.00	0.89	0.79	0.42	-
		0.82	0.47	0.47	0.88	1.00	-
	Bottle	0.85	0.70	0.95	1.00	-	1.00
		0.90	0.89	0.43	0.06	-	0.13
	Cable	0.74	0.93	0.98	0.97	-	-
		0.48	0.18	0.07	0.24	-	-
	Capsule	0.78	1.00	0.96	0.78	-	1.00
		0.43	0.24	0.20	0.03	-	0.03
	Hazelnut	1.00	0.93	0.83	0.90	-	-
		0.07	0.84	0.16	0.07	-	-
	Metal nut	1.00	0.68	0.86	0.55	-	0.32
		0.08	0.77	0.13	0.74	-	0.83
	Pill	0.92	1.00	1.00	0.85	-	1.00
		0.28	0.23	0.24	0.06	-	0.13
	Screw	0.95	0.98	0.41	0.73	-	1.00
		0.06	0.39	0.28	0.13	-	0.10
	Toothbrush	0.75	1.00	1.00	1.00	-	1.00
		0.73	0.97	0.13	0.03	-	0.60
	Transistor	1.00	0.97	0.98	1.00	-	-
		0.03	0.45	0.35	0.15	-	-
	Zipper	1.00	0.97	0.78	0.78	-	-
		0.60	0.63	0.40	0.29	-	-

Hình 3.1 Đối với mỗi danh mục tập dữ liệu, tỷ lệ các mẫu được phân loại chính xác Không có bất thường (hàng trên cùng) và hình ảnh bất thường (dưới cùng hàng) được đưa ra. Phương pháp có giá trị tb cao nhất được tô đậm cho mỗi hàng

Như vậy ta có thể thấy việc áp dụng CNN, CAEs vào việc phát hiện không bất thường hiệu quả hơn.

Tập dữ liệu MVTEC AD có tiềm năng ứng dụng trong nhiều lĩnh vực, bao gồm kiểm tra chất lượng sản phẩm, bảo dưỡng công nghiệp, và phát hiện lỗi tự động trong quá trình sản xuất. Nó cung cấp một bộ kiểm tra quan trọng cho các mô hình phát hiện sự bất thường trong môi trường thực tế. Đây cũng là bước đệm cho sự tối ưu, cải thiện độ chính xác cho các mô hình phát hiện bất thường.

3.1.3 Bài báo số 3

Bài báo “Deep learning for anomaly detection: A survey” [3] nghiên cứu tổng quan về việc ứng dụng học sâu trong lĩnh vực phát hiện bất thường.

Bài báo đã đưa ra các phương thức phát hiện bất thường khác nhau và việc ứng dụng nó vào nhiều lĩnh vực khác nhau như bảo mật mạng, chẩn đoán y học, kiểm tra chất lượng sản phẩm.

1 —Our Survey, 2 —Kwon and Donghwoon Kwon et al. [2017], 5 —John and Derek Ball et al. [2017]
 3 —Kiran and Thomas Kiran et al. [2018], 6 —Mohammadi and Al-Fuqaha Mohammadi et al. [2017]
 4 —Adewumi and Andronicus Adewumi and Akinyelu [2017] 7 —Geert and Kooi et.al Litjens et al. [2017].

		1	2	3	4	5	6	7
Methods	Supervised	✓						
	Unsupervised	✓						
	Hybrid Models	✓						
	one-Class Neural Networks	✓						
Applications	Fraud Detection	✓			✓			
	Cyber-Intrusion Detection	✓	✓					
	Medical Anomaly Detection	✓						✓
	Sensor Networks Anomaly Detection	✓				✓		
	Internet Of Things (IoT) Big-data Anomaly Detection	✓					✓	
	Log-Anomaly Detection	✓						
	Video Surveillance	✓		✓				
	Industrial Damage Detection	✓						

Hình 3.2 So sánh các phương pháp phát hiện bất thường

Bài báo phân loại các phương pháp phát hiện sự bất thường sử dụng học sâu thành các loại chính bao gồm:

Phát hiện dựa trên đặc trưng (Feature-Based Detection): Sử dụng các đặc trưng cụ thể của dữ liệu để phát hiện sự bất thường. Ví dụ là sử dụng mô hình Autoencoder để tái tạo dữ liệu dựa vào đặc trưng và so sánh chúng với dữ liệu gốc.

Phát hiện dựa trên dự đoán (Prediction-Based Detection): Sử dụng mô hình học sâu để dự đoán dữ liệu tiêu chuẩn và sau đó so sánh dự đoán với dữ liệu thực tế. Sự khác biệt giữa dự đoán và dữ liệu thực tế có thể chỉ ra đó là sự bất thường.

Phát hiện dựa trên biểu diễn (Representation-Based Detection): Tạo ra biểu diễn của dữ liệu bằng cách sử dụng mô hình học sâu và sau đó sử dụng các biểu diễn này để phát hiện sự bất thường.

Bài báo cũng đưa ra các khía cạnh khác nhau của phát hiện bất thường gồm unsupervised, semi-supervised, supervised, và giới thiệu 2 kỹ thuật phát hiện bất thường sâu (Deep Anomaly Detection -DAD) là Deep hybrid models và one class neural network. Với mỗi loại kỹ thuật phát hiện bất thường, tác giả đưa ra các khái niệm liên quan đến dữ liệu bình thường, dữ liệu bất thường cùng với các điểm mạnh yếu của nó. Cuối cùng để có thể đánh giá được sự phù hợp của nó đối với sự một tập dữ liệu nhất định.

Bài báo đề cập đến các thách thức trong việc sử dụng học sâu cho phát hiện sự bất thường, bao gồm cách thu thập dữ liệu bất thường, đối phó với sự thay đổi trong dữ liệu, và tối ưu hóa mô hình.

3.1.4 Bài báo số 4

Việc huấn luyện mạng neural cho tác vụ phân loại ảnh thường đòi hỏi một lượng lớn dữ liệu đào tạo, đặc biệt là đối với các kiến trúc mạng neural lớn như Transformers. Tuy nhiên, việc thu thập và gán nhãn cho một số lượng lớn hình ảnh có thể tốn kém và tốn thời gian. Do đó, “Training Vision Transformers with Only 2040 Images” [4] tập trung vào việc huấn luyện Transformers với một lượng hình ảnh đào tạo nhỏ hơn (2040 ảnh).

Với dữ liệu đầu vào số lượng nhỏ, bài báo đã sử dụng một số kỹ thuật để có thể làm cho quá trình huấn luyện với số lượng hình ảnh nhỏ trở nên hiệu quả hơn, như:

Data Augmentation (Tăng cường dữ liệu): Sử dụng các biến đổi dữ liệu để tạo thêm dữ liệu đào tạo giả mạo từ các hình ảnh gốc. Điều này giúp mô hình học được nhiều đặc trưng hơn từ một số lượng ít hình ảnh.

Self-Supervised Learning (Học tự giám sát): Sử dụng một phần của dữ liệu đào tạo để tạo ra nhiệm vụ tự giám sát phụ, trong đó mô hình phải dự đoán các thông tin trong hình ảnh như việc đoán màu sắc hoặc vị trí của đối tượng. Điều này giúp mô hình học được biểu diễn tự nhiên và hữu ích của dữ liệu.

Regularization (Chuẩn hóa): Sử dụng các kỹ thuật chuẩn hóa để ngăn mô hình trở nên quá phức tạp và chống lại hiện tượng overfitting.

Bài báo đã nghiên cứu Vision Transformers (ViTs) được xem như là một giải pháp đang được thay thế mạng nơ-ron tích chập (CNNs), sử dụng bộ dữ liệu nhỏ. Bằng cách kết hợp những kỹ thuật này, bài báo đạt được kết quả tốt trong việc huấn luyện các mô hình Transformers cho phân loại ảnh với lượng dữ liệu đào tạo rất nhỏ. Điều này đề xuất rằng việc sử dụng hiệu quả dữ liệu và kỹ thuật huấn luyện có thể giúp giảm đáng kể yêu cầu về số lượng dữ liệu đào tạo cho các mô hình học máy phức tạp.

3.1.5 Bài báo số 5

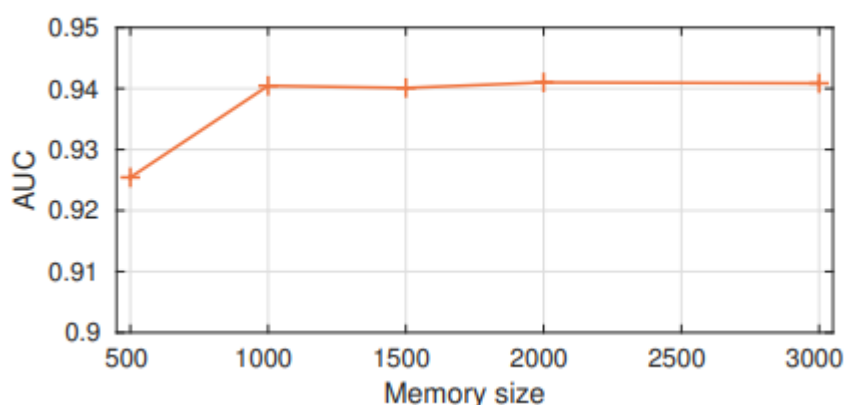
"Memorizing Normality to Detect Anomaly: Memory-augmented Deep Autoencoder for Unsupervised Anomaly Detection" [5] tập trung vào việc phát triển một mô hình autoencoder sâu với tích hợp bộ nhớ để thực hiện việc phát hiện sự bất thường trong dữ liệu mà không cần sử dụng nhãn. Đây là một thách thức lớn trong bài toán phát hiện bất thường.

Để có thể hoàn thành yêu cầu của bài toán, bài báo đã sử dụng một kiến trúc mô hình autoencoder sâu với khả năng ghi nhớ (Memory-augmented Autoencoder-MemAE) để cải thiện hiệu suất của phương pháp phát hiện bất thường không giám sát dựa trên Autoencoder.

"memory-augmented autoencoder" là một biến thể của mô hình autoencoder truyền thống, được thiết kế để cải thiện khả năng của autoencoder trong việc tái tạo và phát hiện sự bất thường trong dữ liệu. Điểm quan trọng của mô hình này là sự tích hợp của một bộ nhớ bổ sung, giúp mô hình ghi nhớ thông tin quan trọng từ dữ liệu đào tạo để sử dụng trong quá trình đánh giá sự tương tự và phát hiện sự bất thường trong dữ liệu mới. Thông tin quan trọng này có thể là các đặc điểm, mẫu dữ liệu, hoặc biểu diễn của phần bình thường của tập dữ liệu.

Khi một điểm dữ liệu mới được đưa vào mô hình, nó sẽ được so sánh với thông tin trong bộ nhớ để xác định sự tương tự. Nếu sự khác biệt lớn, điểm dữ liệu này có thể được coi là một sự bất thường.

Tác giả triển khai với learning rate 0,0001, sử dụng PyTorch và hàm tối ưu hóa Adam. Sau khi tiến hành các thí nghiệm bằng cách sử dụng khác nhau cài đặt kích thước bộ nhớ và hiển thị các giá trị AUC, với các giá trị bộ nhớ đủ lớn MemAE có thể tạo ra kết quả hợp lý.



Hình 3.3 Giá trị AUC khi thử nghiệm phương pháp MemAE

AUC là viết tắt của "Area Under the Receiver Operating Characteristic Curve." Nó là một số đo hiệu năng thường được sử dụng trong học máy và học sâu để đánh giá khả năng của mô hình phân loại (classification model). Giá trị AUC nằm trong khoảng từ 0 đến 1, với giá trị lớn hơn cho thấy mô hình có hiệu suất tốt hơn trong việc phân loại. Nếu AUC bằng 0.5, điều đó tương đương với việc mô hình phân loại một cách ngẫu nhiên.

3.1.6 Bài báo số 6

"Deep learning for anomaly detection: A review" [6] đã nêu lên được sự phức tạp và những thách thức chưa được giải quyết trong phát hiện sự bất thường trong nhiều lĩnh vực. Ví dụ như những trường hợp hành vi đột ngột, cấu trúc dữ liệu và phân phối, chẳng ai có thể biết được cho đến khi trên thực tế xảy ra, chẳng hạn như khủng bố, xâm nhập mạng, gian lận. Các bất thường là các trường hợp tương phản với điều bình thường, chiếm tỉ lệ cực ít, vì thế nếu không nói là không thể thu thập một lượng lớn các trường hợp bất thường được dán nhãn.

Hiện nay có một vài phương pháp có thể khắc phục được phần lớn các khuyết điểm đó là sử dụng mạng lưới thần kinh sâu (Deep neural networks) như multilayer perceptron (MLP) networks, convolutional neural networks (CNN), recurrent neural networks (RNN). Và mạng autoencoder là kỹ thuật thường được sử dụng, bởi nó gồm một mạng mã hóa, một giải mã. Thông tin được lưu giữ lại là thông tin có nhiều đặc trưng nhất. Cũng như các tác giả khác, bài báo đưa ra các phương pháp học máy là supervised, semi-supervised, self-supervised và unsupervised.

Bài báo đánh giá ưu điểm và nhược điểm của việc sử dụng deep learning trong phát hiện sự bất thường. Trong số những ưu điểm, học sâu có khả năng học các biểu diễn cấp cao của dữ liệu và thích nghi với nhiều loại dữ liệu khác nhau. Tuy nhiên, nó cũng đề cập đến nhược điểm như đòi hỏi lượng dữ liệu lớn và khả năng khai thác mô hình.

3.2 Nghiên cứu điển hình

Sau khi đã tìm hiểu được một vài phương pháp chung mà những bài báo lớn đã nêu ra, đã có được một số kiến thức cơ bản để có thể tìm hiểu về một phương

pháp cụ thể được giới thiệu trong bài báo “CutPaste: Self-Supervised Learning for Anomaly Detection and Localization” [7]. Đây là bài báo tương đối mới, có kết quả độ chính xác rất cao trong việc phát hiện bất thường, nên em chọn bài báo này để nghiên cứu kỹ.

Bài báo giới thiệu một phương pháp mới trong lĩnh vực xử lý hình ảnh, được gọi là "CutPaste". Phương pháp này sử dụng học tự giám sát (self-supervised learning) để phát hiện và xác định vị trí các điểm bất thường trong dữ liệu hình ảnh. Cách tiếp cận này không đòi hỏi dữ liệu hình ảnh được gắn nhãn, cho phép mô hình học máy tự động phân biệt điểm bất thường và xác định vị trí chúng trong hình ảnh.

Phương pháp "CutPaste" bao gồm các bước sau:

Cắt và Dán: Ở bước này, một phần của hình ảnh gốc được cắt ra và sau đó dán lại vào một vị trí ngẫu nhiên trong hình ảnh. Việc cắt và dán này tạo ra các biến thể của ảnh gốc, trong đó một phần của ảnh có thể bị biến đổi hoặc là điểm bất thường. Có các kiểu CutPaste như:

- Cutpaste-Scar: là sử dụng một hình chữ nhật có dải màu bất kỳ rồi dán nó lên vật thể
- CutMix: là cắt một hình ảnh chữ nhật từ một hình ảnh và dán nó vào vị trí ngẫu nhiên của một hình ảnh khác

Các thí nghiệm trong báo cáo hầu hết được sử dụng trên tập dữ liệu MVTec Anomaly Detection, bao gồm 10 loại đối tượng và 5 loại vật liệu.

Category		DOCC [45]	U-Student [6]	P-SVDD [61]	Rotation	Cutout	Scar	CutPaste	CutPaste (scar)	CutPaste (3-way)	Ensemble
texture	carpet	90.6	95.3	92.9	29.7±1.4	35.3±2.3	92.7±0.4	67.9±1.8	94.6±0.6	93.1±1.1	93.9
	grid	52.4	98.7	94.6	60.5±7.0	57.5±3.0	74.4±2.5	99.9 ±0.1	95.5±0.3	99.9 ±0.1	100.0
	leather	78.3	93.4	90.9	55.2±1.4	67.7±1.5	99.9 ±0.1	99.7±0.1	100.0 ±0.0	100.0 ±0.0	100.0
	tile	96.5	95.8	97.8	70.1±1.9	71.8±4.0	96.7 ±0.9	95.9 ±1.0	89.4±2.8	93.4±1.0	94.6
	wood	91.6	95.5	96.5	95.8±1.1	92.0±0.8	98.9 ±0.2	94.9±0.5	98.7 ±0.3	98.6 ±0.5	99.1
	average	81.9	95.7	94.5	62.3±2.6	64.9±2.3	92.5±0.8	91.7±0.7	95.7±0.8	97.0 ±0.5	97.5
object	bottle	99.6	96.7	98.6	95.0±0.7	88.7±0.8	98.5±0.2	99.2±0.2	98.0±0.5	98.3±0.5	98.2
	cable	90.9	82.3	90.3	85.3±0.8	80.2±1.4	78.3±1.7	87.1±0.8	78.8±2.9	80.6±0.5	81.2
	capsule	91.0	92.8	76.7	71.8±1.4	69.5±1.1	82.9±0.7	87.9±0.7	95.3 ±0.8	96.2 ±0.5	98.2
	hazelnut	95.0	91.4	92.0	83.6±0.8	69.7±1.3	98.9 ±0.2	91.3±0.6	96.7±0.4	97.3±0.3	98.3
	metal nut	85.2	94.0	94.0	72.7±0.5	84.6±0.7	86.9±1.5	96.8±0.5	97.9±0.2	99.3 ±0.2	99.9
	pill	80.4	86.7	86.1	79.2±1.4	78.7±0.7	82.2±1.4	93.4 ±0.9	85.8±1.3	92.4 ±1.3	94.9
	screw	86.9	87.4	81.3	35.8±2.9	17.6±4.4	11.3±2.2	54.4±1.7	83.7±0.7	86.3±1.0	88.7
	toothbrush	96.4	98.6	100.0	99.1±0.2	98.1±0.6	94.8±1.0	99.2±0.2	96.7±0.4	98.3±0.9	99.4
	transistor	90.8	83.6	91.5	88.9±0.4	82.5±1.2	92.0±0.7	96.4 ±0.7	91.1±0.6	95.5 ±0.5	96.1
	zipper	92.4	95.8	97.9	74.3±1.6	75.7±1.0	86.8±0.9	99.4 ±0.1	99.5 ±0.1	99.4 ±0.2	99.9
	average	90.9	90.9	90.8	78.6±1.1	74.5±1.3	81.3±1.1	90.5±0.6	92.4±0.8	94.3 ±0.6	95.5
average		87.9	92.5	92.1	73.1±1.6	71.3±1.6	85.0±1.0	90.9±0.7	93.5±0.8	95.2 ±0.6	96.1

Hình 3.4 So sánh giá trị trung bình AUC với các phương pháp và các tập dữ liệu

Bài báo đã thực hiện thí nghiệm 5 lần với các đối tượng ngẫu nhiên và cũng báo cáo giá trị trung bình của chỉ số AUC cho các loại vật liệu, đối tượng và tất cả các loại ghi lại chỉ số trung bình AUC.

Nhóm tác giả so sánh với phương pháp deep one -class classifier (DOCC), uninformed student và P-SVDD. Tuy nhiên kết quả ở phương pháp biến đổi dữ liệu bằng cách xoay không đạt kết quả tốt với AUC là 73.1 trong việc phát hiện khiếm khuyết so với phương pháp Scar-1 biến thể của Cutout (85.0).

CutPaste và CutPaste-scar cải thiện dự đoán cutpaste và scar bằng cách tránh các giải pháp thụ động (là cần có nhãn, còn cutpaste học tập không giám sát), đạt hiệu suất tốt hơn so với các phương pháp khác với các chỉ số AUC lần lượt là

90.9 và 93.5. Với CutPaste 3-way (thay đổi các giá trị siêu tham số tối ưu nhất) thì đã đạt được giá trị tốt nhất là 95.2

Bài báo cũng nghiên cứu tác động của các siêu tham số của CutPaste, như learning rate, epoch, Tất cả các thí nghiệm được thực hiện trên cài đặt CutPaste 3-way. Dựa trên dataset của MVTEC AD, cùng với các siêu tham số (hyperparameter) như sau:

Learning rates $\in \{0.1, \mathbf{0.03}, 0.01, 0.003\}$. Number of training epochs $\in \{128, 192, 256, 320, 384\}$. Maximum jitter intensity on patch $\in \{0, \mathbf{0.1}, 0.2, 0.3\}$. Maximum size of patch $\in \{0.1, \mathbf{0.15}, 0.2, 0.3\}$. Với các tham số mặc định thử nghiệm thì được tô màu đỏ.

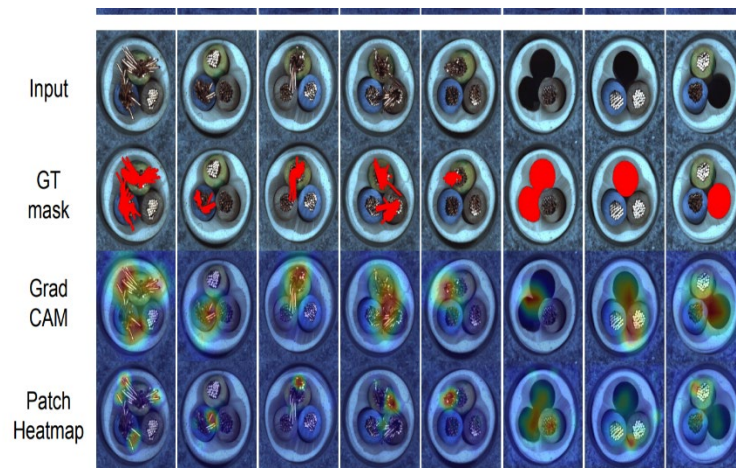
Category	Learning rates				Number of epochs				
	0.1	0.03	0.01	0.003	128	192	256	320	384
texture	97.1 \pm 0.3	97.0 \pm 0.5	97.2 \pm 0.3	96.1 \pm 0.7	96.6 \pm 0.4	96.1 \pm 0.7	97.0 \pm 0.5	97.0 \pm 0.4	96.3 \pm 0.4
object	94.4 \pm 0.6	94.3 \pm 0.6	94.2 \pm 0.6	93.9 \pm 0.5	94.9 \pm 0.6	94.5 \pm 0.4	94.3 \pm 0.6	94.7 \pm 0.5	94.0 \pm 0.6
all	95.3 \pm 0.5	95.2 \pm 0.6	95.2 \pm 0.5	94.6 \pm 0.6	95.4 \pm 0.5	95.0 \pm 0.5	95.2 \pm 0.6	95.5 \pm 0.4	94.8 \pm 0.5

Hình 3.5 AUC khi thay đổi siêu tham số

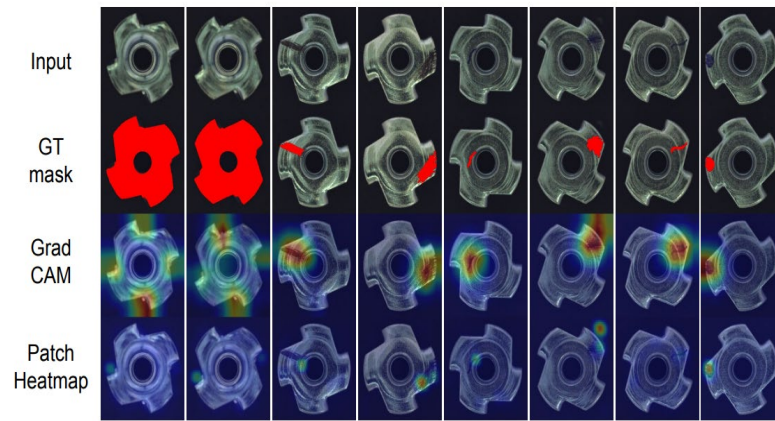
Category	Jitter intensity				Size of patch			
	0.0	0.1	0.2	0.3	0.1	0.15	0.2	0.3
texture	96.2 \pm 0.6	97.0 \pm 0.5	97.4 \pm 0.3	97.5 \pm 0.2	97.1 \pm 0.4	97.0 \pm 0.5	95.8 \pm 0.9	96.6 \pm 0.4
object	94.3 \pm 0.6	94.3 \pm 0.6	94.5 \pm 0.5	94.5 \pm 0.5	94.5 \pm 0.5	94.3 \pm 0.6	94.4 \pm 0.5	94.5 \pm 0.5
all	94.9 \pm 0.6	95.2 \pm 0.6	95.5 \pm 0.4	95.5 \pm 0.4	95.3 \pm 0.5	95.2 \pm 0.6	94.8 \pm 0.6	95.2 \pm 0.5

Hình 3.6 AUC khi thay đổi siêu tham số 2

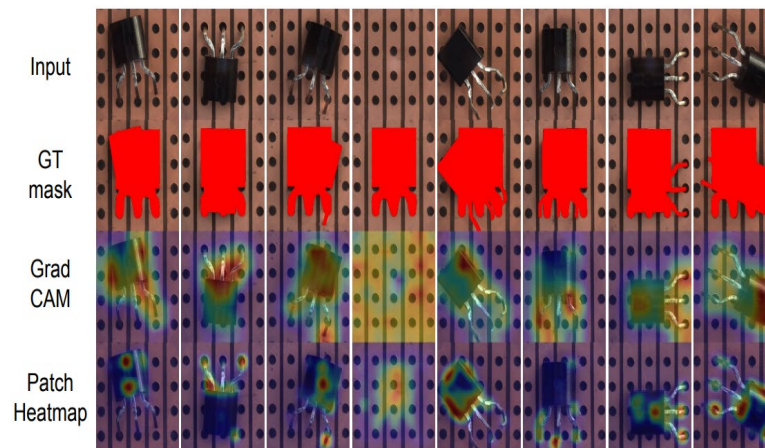
Có thể thấy rằng phương pháp cutpaste khá ổn định với các tham số learning rate và number epoch khác nhau. Tuy nhiên vẫn có một vài trường hợp thất bại.



Hình 3.7 Điểm bất thường ở lỗi cấp



Hình 3.8 Điểm bất thường ở đai ốc



Hình 3.9 Điểm bất thường ở transistor

- Lỗi cáp thiếu thành phần ở cột 6,7,8
- Đai ốc lật chiều ở cột 1,2
- Transistor bị lệch hoặc thiếu hàng

Với một số trường hợp dị thường thì model vẫn chưa thể hoàn thiện được, nhưng có thể nói rằng độ chính xác cao hơn so với các phương pháp khác như DOCC, P-SVDD.

Phương pháp "CutPaste" có nhiều ứng dụng quan trọng trong lĩnh vực xử lý hình ảnh và phát hiện bất thường, bao gồm:

Y tế: Phát hiện các điểm bất thường trong hình ảnh y tế, chẳng hạn như xác định vị trí của các khối u trong hình ảnh chụp MRI hoặc CT.

Kiểm tra chất lượng sản phẩm: Kiểm tra chất lượng sản phẩm công nghiệp, như xác định các khuyết tật trên bề mặt sản phẩm

An ninh và giám sát: Dùng để phát hiện các sự kiện bất thường trong hình ảnh từ camera an ninh, ví dụ như phát hiện người lạ trong khu vực cấm

Bài báo "CutPaste: Self-Supervised Learning for Anomaly Detection and Localization" [7] đã đưa ra phương pháp CutPaste và nêu lên được các ưu điểm cũng như một số nhược điểm còn tồn tại. Phương pháp này có nhiều ứng dụng quan trọng và hứa hẹn trong lĩnh vực xử lý hình ảnh và xác định bất thường.

CHƯƠNG 4. TRIỂN KHAI THUẬT TOÁN AUTOENCODER TRÊN TẬP DỮ LIỆU NHỎ CỦA MVCTEC

4.1 Bộ dữ liệu

Dữ liệu được cung cấp bởi MVTec Anomaly Detection, bộ dữ liệu bao gồm 480 hình ảnh với độ phân giải 1024x1024 về ốc vít, trong đó tập huấn luyện gồm 320 hình ảnh ốc vít không có bất thường và tập thử nghiệm bao gồm 120 hình ảnh ốc vít bất thường và 40 hình ảnh ốc vít không bất thường.

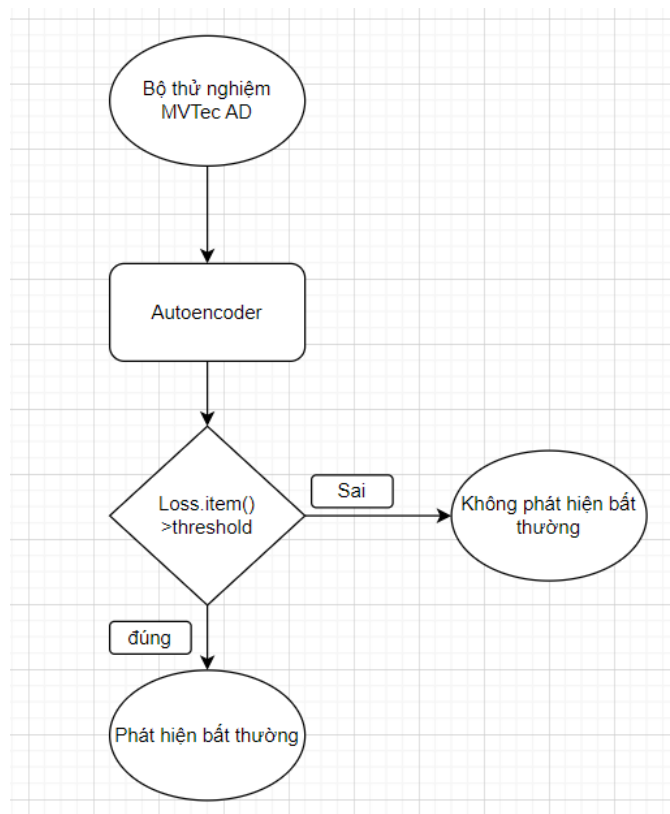
4.2 Phương pháp sử dụng

Lựa chọn mạng kiến trúc nơ ron sâu CNN, mạng autoencoder để triển khai bài toán làm quen và với các hyperparameter như sau:

- Learning rate= 0.001
- Number_epoch_train=80
- Stride =2, padding =1
- Hàm kích hoạt ReLU, hàm tối ưu Adam

Bởi vì số lần epoch train khá lớn nên mất nhiều thời gian mỗi lần chạy, gây ra hiện tượng nóng máy, thời gian lâu. Để có thể tối ưu thời gian và tránh hiện tượng quá tải cho laptop, sử dụng phương pháp lưu và tải model.

4.3 Xây dựng thuật toán và kiểm thử

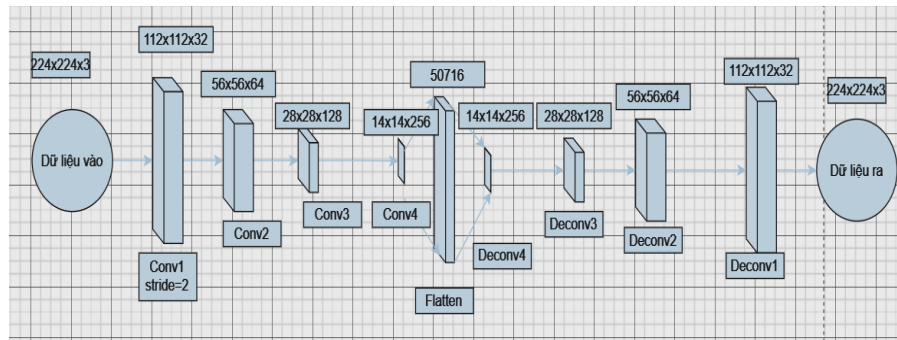


Hình 4.1 Lưu đồ thuật toán phát hiện bất thường bằng autoencoder

Từ lưu đồ thuật toán, đưa bộ dữ liệu nhỏ của MVTec vào mô hình Autoencoder, mô hình sẽ thực hiện tính toán mất mát so với các dữ liệu hoàn chỉnh khi học tập.

Từ đó mô hình so sánh với giá trị “threshold” để đưa ra kết luận việc hình ảnh đưa vào có bất thường hay không.

4.3.1 Mô hình autoencoder

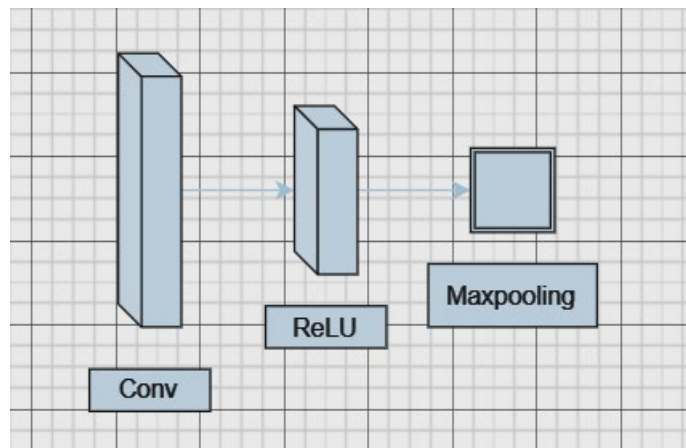


Hình 4.2 Mô hình autoencoder

Các hình ảnh đầu vào được chuyển đổi để phù hợp với mô hình và đạt được kết quả tốt nhất. Mô hình gồm 04 lớp tích chập để thu thập các đặc trưng của hình ảnh đầu vào. Với mỗi đầu ra của lớp tích chập này là đầu vào của lớp tích chập tiếp theo, cho đến khi thu được các đặc trưng tốt nhất.

Để rồi các đặc trưng được đưa vào bộ giải mã trong mô hình để tái tạo lại hình ảnh, từ đó so sánh với các hình ảnh “bình thường” lúc học để tính toán mất mát.

Sau khi đi qua mỗi lớp tích chập thì các đầu ra sẽ được chuyển vào hàm ReLU và MaxPooling để giảm kích thước của mỗi hình ảnh mà vẫn giữ được các đặc trưng.



Hình 4.3 Trước khi vào các lớp tích chập khác, hình ảnh được đưa qua các hàm giảm kích thước

4.4 Kiểm tra và đánh giá

Để đánh giá một mô hình, ta cần xác định các thông số sau đây:

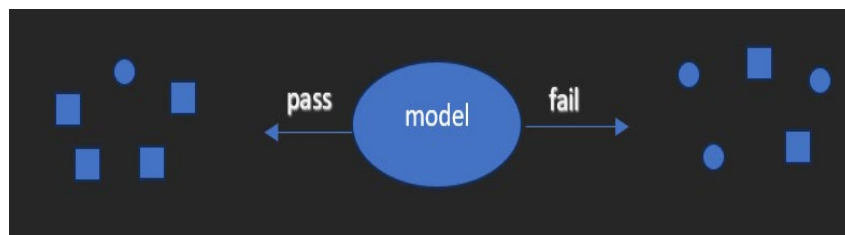
- Confusion Matrix (Ma trận nhầm lẫn): Ma trận này thể hiện sự phân loại của mô hình trên mỗi lớp. Là một bảng 2x2 hoặc $n \times n$ (với n là số lớp)
- Accuracy (Độ chính xác): Là tỷ lệ giữa số lượng dự đoán đúng và tổng số điểm dữ liệu. Công thức: $(TP + TN) / (TP + TN + FP + FN)$.
- Precision (Độ chính xác dương tính): Là tỷ lệ giữa số lượng dự đoán đúng là dương tính (True Positive) và tổng số dự đoán dương tính (True

Positive + False Positive). Precision thể hiện khả năng của mô hình dự đoán đúng khi nó dự đoán là positive. Công thức: $TP / (TP + FP)$.

Recall (Tỷ lệ phục hồi hoặc True Positive Rate): Là tỷ lệ giữa số lượng dự đoán đúng là dương tính (True Positive) và tổng số thực tế là dương tính (True Positive + False Negative). Recall thể hiện khả năng của mô hình dự đoán đúng các trường hợp positive. Công thức: $TP / (TP + FN)$.

F1 Score: Là sự kết hợp của Precision và Recall để đánh giá mô hình dựa trên cả hai chỉ số. F1 Score thường được sử dụng khi cần cân nhắc cả Precision và Recall trong bài toán phân loại. Công thức: $2 * (Precision * Recall) / (Precision + Recall)$.

Để dễ hiểu ta lấy ví dụ về bài toán một học sinh dự đoán mình có vượt qua bài kiểm tra cuối kỳ hay không với quy ước, qua sẽ là pass tương ứng hình vuông, còn không qua là fail tương ứng với hình tròn.



Hình 4.4 Ví dụ về các thông số

TP (khi bạn đoán là pass và kết quả dự đoán này đúng): 4

TN (khi bạn đoán là fail và kết quả dự đoán này đúng): 3

FP (khi bạn đoán là pass nhưng kết quả là “fail”): 1

FN (khi bạn đoán là fail nhưng kết quả là “pass”): 2

Quay trở lại bài toán phát hiện bất thường ở đỉnh vít, ta có:

TP: 40, TN: 108, FP: 12, FN: 0

Confusion Matrix:

40	12
0	108

Accuracy = 92.50

Precision = 76.92

Recall = 100

F1 Score = 86.95

Nếu mục tiêu là tối ưu hóa độ chính xác tổng thể, có thể chọn mô hình với accuracy cao nhất.

Nếu quan trọng việc giảm thiểu các sai lầm dương tính (FP), có thể tập trung vào precision.

Nếu quan trọng việc bắt được tất cả các trường hợp tích cực và có thể chấp nhận một số dự đoán sai (FN), tập trung vào recall.

Nếu muốn cân nhắc cả precision và recall, có thể sử dụng F1 Score.

Đánh giá

Mô hình vẫn đang có một số sai sót nhất định, cần khắc phục

Cần tăng số lượng data đầu vào để train mô hình một cách chính xác hơn

Mô hình có thể đang bị overfitting, để tránh overfitting ta cần làm:

- a. Tăng số lượng hình ảnh cần phân loại lên
- b. Sử dụng tập dữ liệu chưa từng dùng để kiểm tra
- c. Điều chỉnh các siêu tham số (hyperparameter) phù hợp
- d. Theo dõi mô hình thường xuyên để sớm phát hiện overfitting càng sớm
- e. Giảm độ phức tạp của model

CHƯƠNG 5. CHƯƠNG KẾT LUẬN

5.1 Đánh giá mức độ hoàn thiện

So với các nội dung công việc đề ra như dự kiến, đồ án đã thực hiện được các nội dung như sau:

STT	Tên công việc	Thời gian thực tế	Kết quả đạt được
1	Tìm hiểu tổng quan học máy (ML), học sâu (DL), đặc biệt	01/08-08/08 (1 tuần)	Tìm hiểu tổng quan, thuật toán của học sâu, các vấn đề triển khai CNN, Autoencoder, ứng dụng transformer
2	Tìm và đọc một số bài báo tổng quan về phát hiện bất thường sử dụng hình ảnh	09/08-23/08 (2 tuần)	Đọc các bài báo: <ul style="list-style-type: none">- A Survey of Visual Sensory Anomaly Detection- MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection- Deep learning for anomaly detection: A survey- Training Vision Transformers with Only 2040 Images- Memorizing Normality to Detect Anomaly: Memory-augmented Deep Autoencoder for Unsupervised Anomaly Detection- Deep learning for anomaly detection: A review
3	Tìm hiểu nghiên cứu điển hình về bài toán phát hiện bất thường	23/08-08/09 (2 tuần)	Đọc bài báo: CutPaste: Self-Supervised Learning for Anomaly Detection and Localization, tìm hiểu cách hoạt động của phương pháp CutPaste để tạo ra các bất thường, so sánh các kết quả với những nghiên cứu đã có.
4	Triển khai một bài toán phát hiện bất thường	08/09-29/09 (3 tuần)	Sử dụng mạng nơ-ron tích chập, áp dụng autoencoder để thử nghiệm phát hiện bất thường trên tập dữ liệu MVTec AD
5	Viết báo cáo	29/09-05/10 (1 tuần)	Kiểm tra lại tổng quan đề tài và viết báo cáo

Bảng 2 Bảng kết quả thực hiện

Kết quả thực hiện: Tồn tại một số bất cập như là về giai đoạn tìm hiểu tổng quan các bài báo là chưa thể tóm gọn nội dung nghiên cứu một cách nhanh chóng, dẫn đến thời gian dự kiến kéo dài thêm 1 tuần, giai đoạn nghiên cứu bài báo cụ thể bị hạn chế về kiến thức chuyên môn và giai đoạn triển khai bài toán phát hiện bất thường phát sinh các vấn đề cần giải quyết như tối ưu hóa model, lưu các model và cải thiện độ chính xác của bài toán. Tổng kết mức độ hoàn thành ở các giai đoạn có thể ở mức tạm chấp nhận, nhờ sự hỗ trợ cũng như góp ý của thầy Văn nên các sự cố phát sinh không gây ra ảnh hưởng quá lớn đến đề án.

5.2 Hướng phát triển trong tương lai

Bên cạnh những thành công bước đầu vẫn không thể tránh khỏi những hạn chế về nhiệm vụ phát hiện bất thường thông qua hình ảnh. Nguyên nhân chủ yếu do quá hạn chế về thời gian và đây cũng là một vấn đề hoàn toàn mới đối với bản thân, em mong trong thời gian sắp tới có thể khắc phục tốt những mặt hạn chế để hoàn thành sản phẩm một cách toàn diện hơn. Em đưa ra những hướng phát triển của đề tài trong tương lai:

- Cải thiện độ chính xác để giảm thiểu những sai sót.
- Có thể phát hiện bất thường ở thời gian thực thông qua camera.
- Nhận dạng được nhiều đối tượng hơn.

TÀI LIỆU THAM KHẢO

- [1] G. X. J. W. Y. L. C. W. F. Z. a. Y. J. Xi Jiang, "A Survey of Visual Sensory Anomaly Detection," 2022.
- [2] M. F. D. S. a. C. S. Paul Bergmann, "MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition," 2019.
- [3] R. C. a. S.-j. Chawla, "Deep learning for anomaly detection: A survey. arXiv preprint arXiv," 2019.
- [4] H. Y. a. J. W. Yunhao Cao, " Training vision transformers with only 2040 images.," 2022.
- [5] L. L. V. L. B. S. M. R. M. S. V. a. A. v. d. H. Dong Gong, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," 2019.
- [6] C. S. L. C. a. A. V. D. H. Guansong Pang, "Deep learning for anomaly detection: A review. ACM Computing Surveys (CSUR)," 2021.
- [7] K. S. J. Y. T. P. Chun-Liang Li, "CutPaste: Self-Supervised Learning for Anomaly Detection and Localization. In Proceedings of the IEEE/CV," 2021.
- [8] P. L. H. B. Y. & M. P. A. Vincent, Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning, 2008.
- [9] J. X. L. & C. E. Xie, Image denoising and inpainting with deep neural networks. In Advances in neural information processing systems, 2012.
- [10] P. & M. J. Perona, Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on pattern analysis and machine intelligence, 1990.
- [11] L. Gondara, Medical image denoising using convolutional denoising autoencoders. In 2016 IEEE 16th international conference on data mining workshops (ICDMW) IEEE, 2016.
- [12] R. R. & D. D. L. Coifman, Translation-invariant de-noising. In Wavelets and statistics, Springer, New York, 1995.
- [13] I. D. a. A. Team, AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the difference?.AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the difference?, 2023.
- [14] A. Alderfer, Anomaly Detection with Autoencoders.

- DrewAlderfer/anomaly_detection at working, 2023.
- [15] T. c. k. h. Đ. h. Đ. Á, Ứng dụng kỹ thuật bộ mã tự động tích chập (CONVOLUTIONAL AUTOENCODER) trong khử nhiễu hình ảnh, 2022.
- [16] M. Software, MVTec Anomaly Detection Dataset.
- [17] K. B. M. F. D. S. C. S. Paul Bergmann, "The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection; in: International Journal of Computer Vision," 2021.
- [18] Y. Z. Q. Z. D. X. S. P. a. H. Z. Jinlei Hou, "Divide-and-assemble: Learning block-wise memory for unsupervised anomaly detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision," 2021.
- [19] U. M. D. C. a. J. S. Jonathan Masci, "Stacked convolutional auto-encoders for hierarchical feature extraction. In ICANN," 2011.
- [20] <https://glints.com/vn/blog/machine-learning-la-gi/>, Ngày truy cập lần cuối 28/09/2023.
- [21] <https://viblo.asia/p/machine-learning-tong-quan-ve-machine-learning-RQqKLxaOK7z>, ngày truy cập lần cuối 19/09/2023.
- [22] <https://viblo.asia/p/tan-man-ve-generative-models-part-1-cac-mo-hinh-autoencoder-vaes-4P856rw35Y3>, ngày truy cập lần cuối 19/09/2023.
- [23] <https://nttuan8.com/bai-6-convolutional-neural-network/>, Ngày truy cập lần cuối 23/09/2023.

