ĐẠI HỌC HUẾ

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

BỘ MÔN KHOA HỌC DỮ LIỆU VÀ TRÍ TUỆ NHÂN TẠO

HOÀNG NỮ THU PHƯƠNG – 21E1010003

ỨNG DỤNG THỊ GIÁC MÁY TÍNH VÀ HỌC SÂU VÀO KIỂM TRA CHẤT LƯỢNG LON RỖNG TRONG DÂY CHUYỀN CÔNG NGHIỆP

KHÓA LUẬN TỐT NGHIỆP

KHOA HỌC DỮ LIỆU VÀ TRÍ TUỆ NHÂN TẠO

THÀNH PHỐ HUẾ, NĂM 2025

HOÀNG NỮ THU PHƯƠNG – KHÓA LUẬN TỐT NGHIỆP – KHDL&TTNT –HUẾ 2025

ĐẠI HỌC HUẾ

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

BỘ MÔN KHOA HỌC DỮ LIỆU VÀ TRÍ TUỆ NHÂN TẠO

HOÀNG NỮ THU PHƯƠNG – 21E1010003

ỨNG DỤNG THỊ GIÁC MÁY TÍNH VÀ HỌC SÂU VÀO KIỂM TRA CHẤT LƯỢNG LON RỖNG TRONG DÂY CHUYỀN CÔNG NGHIỆP

KHÓA LUẬN TỐT NGHIỆP

KHOA HỌC DỮ LIỆU VÀ TRÍ TUỆ NHÂN TẠO

Giảng viên hướng dẫn:

TS. Nguyễn Thị Hà Phương

THÀNH PHỐ HUẾ, NĂM 2025

HOÀNG NỮ THU PHƯƠNG – KHÓA LUẬN TỐT NGHIỆP – KHDL&TTNT –HUẾ 2025

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của Tiến sĩ Nguyễn Thị Hà Phương. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Khóa luận tốt nghiệp còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Khóa luận tốt nghiệp của mình. Khoa Kỹ thuật và Công nghệ - Đại học Huế không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

Thành phố Huế, ngày 12 tháng 12 năm 2025.

(Tác giả luận văn ký ghi rõ họ tên)

Hoàng Nữ Thu Phương

LỜI CẢM ƠN

Để hoàn thành Khóa luận tốt nghiệp này, em xin chân thành cảm ơn Khoa Kỹ thuật và Công nghệ, Đại học Huế đã luôn tạo điều kiện thuận lợi cho sự nghiệp học tập của sinh viên. Những kiến thức được học không chỉ là nền tảng vững chắc mà còn là hành trang quý báu cho tương lai.

Em xin gửi lời cảm ơn sâu sắc đến thầy Trưởng khoa, PGS.TS. Nguyễn Quang Lịch, người luôn dành sự quan tâm lớn lao đến sinh viên, tạo ra môi trường học tập hiện đại, năng động và giàu tính ứng dụng.

Đặc biệt, em xin cảm ơn cô TS. Nguyễn Thị Hà Phương đã tận tình hướng dẫn, giúp đỡ em trong suốt quá trình thực hiện Khóa luận. Những kiến thức và kinh nghiệm cô truyền đạt đã giúp em có đủ hành trang và động lực để hoàn thiện công trình này.

Em cũng xin cảm ơn quý thầy cô, cán bộ giảng viên trong Khoa đã hết lòng truyền đạt tri thức, đồng thời gửi lời cảm ơn đến Công ty TNHH Bia Carlsberg Việt Nam, nơi em có cơ hội thực tập và trải nghiệm thực tế. Những hỗ trợ từ các anh chị trong công ty đã giúp em tích lũy thêm nhiều kiến thức và kỹ năng bổ ích, góp phần quan trọng để hoàn thành Khóa luận này.

Cuối cùng, em xin gửi lời cảm ơn sâu sắc đến gia đình và bạn bè, những người luôn động viên, ủng hộ và là nguồn động lực lớn để em vượt qua khó khăn. Nhận thức rõ bản thân còn có nhiều hạn chế về chuyên môn, em rất mong nhận được sự góp ý và chỉ bảo thêm của quý thầy cô để Khóa luận được hoàn thiện hơn.

Thành phố Huế, ngày 12 tháng 12 năm 2025.

Hoàng Nữ Thu Phương

TÓM TẮT

Đề tài "Ứng dụng thị giác máy tính và học sâu vào kiểm tra chất lượng lon rỗng trong dây chuyền công nghiệp" tập trung nghiên cứu, thực nghiệm và so sánh hiệu năng của ba nhóm kiến trúc mạng nơ-ron tích chập (CNN) nhằm tự động hóa quy trình kiểm soát chất lượng lon rỗng. Nghiên cứu được triển khai thông qua việc so sánh các phương pháp tiếp cận khác nhau trên cùng một bộ dữ liệu cân bằng.

Autoencoder (AE): Áp dụng phương pháp Học không giám sát để phát hiện bất thường (Anomaly Detection). Mô hình đã sử dụng chỉ số Youden's J statistic để thiết lập ngưỡng tối ưu. Tuy nhiên, hiệu suất phát hiện lỗi NG còn hạn chế do đặc thù không huấn luyện trên dữ liệu lỗi, dẫn đến việc bỏ sót một số khiếm khuyết.

EfficientNetB0: Sử dụng kỹ thuật Học chuyển giao (Transfer Learning) cho bài toán phân loại nhị phân (Binary Classification). Mô hình đạt độ nhạy cao trong việc nhận diện sản phẩm đạt chuẩn OK, nhưng đối mặt với rủi ro đáng kể về dương tính giả (False Positive), khiến mô hình bị lỗi nghiêm trọng.

Mô hình YOLOv8-seg và YOLOv11-seg: Áp dụng phương pháp phát hiện và phân đoạn đối tượng (Instance Segmentation). Mô hình YOLOv8-seg đã chứng minh khả năng phân đoạn và phân loại vượt trội nhất, đạt độ chính xác cao nhất ở cả lớp OK (99.8%) và lớp NG (79.0%) trên tập kiểm thử độc lập. Mặc dù YOLOv11-seg, với sự tích hợp của cơ chế chú ý không gian nâng cao (Spatial Attention Mechanisms) nhưng kết quả thu được lại không đạt kì vọng.

Khóa luận kết luận YOLOv8-seg là giải pháp có tiềm năng cho việc định vị lỗi. Tuy nhiên, để đáp ứng yêu cầu khắt khe của môi trường sản xuất, cần phải tối ưu hóa mô hình nhằm cân bằng hiệu suất giữa Precision và Recall cho cả hai lớp để giảm thiểu việc phân loại sai tính chất của lon hoặc kết hợp nhiều mô hình với nhau để xây dựng hệ thống kiểm tra đa tầng đảm bảo hạn chế tối đa nhận diện lỗi.

MỤC LỤC

LỜI CAM ĐOAN i
LỜI CẢM ƠN ii
TÓM TẮT iii
MỤC LỤC iv
DANH MỤC CÁC CHỮ VIẾT TẮT VÀ KÝ HIỆU vii
DANH MỤC CÁC BẢNG BIỂU viii
DANH MỤC CÁC SƠ ĐỒ, ĐỒ THỊ, HÌNH VẼ ix
PHẦN I. PHẦN MỞ ĐẦU 1
PHẦN II. TỔNG QUAN VẤN ĐỀ NGHIÊN CỨU 5
PHẦN III. ĐỐI TƯỢNG, PHẠM VI VÀ PHƯƠNG PHÁP NGHIÊN CỨU 7
PHẦN IV. NỘI DUNG VÀ KẾT QUẢ NGHIÊN CỨU 10
Chương 1. Tổng quan và cơ sở lý thuyết 10
1.1. Tổng quan về Thị giác máy tính trong Công nghiệp ***10***
1.1.1. Vai trò của xử lý ảnh trong kiểm soát chất lượng (Quality Control) *10*
1.1.2. Các bài toán cơ bản trong kiểm tra bề mặt sản phẩm *11*
1.2. Tổng quan về học máy ***12***
1.2.1. Thế nào là học máy *12*
1.2.2. Phân loại học máy *13*
1.2.3. Học sâu *16*
1.3. Mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) ***17***
1.3.1. Kiến trúc cơ bản của mạng nơ-ron tích chập *18*
1.3.2. Các hàm kích hoạt phi tuyến (Activation Functions) *23*
1.3.3. Hàm mất mát (Loss Functions) *24*
1.4. Kiến trúc Autoencoder (AE) ***25***
1.4.1. Kiến trúc cơ bản *25*
1.4.2. Cơ chế phát hiện bất thường *25*

1.5. Mô hình EfficientNet ***26***
1.5.1. Nguyên tắc Compound Scaling *26*
1.5.2. Ứng dụng học chuyển giao (Transfer Learning) *26*
1.6. Mô hình YOLO (You Only Look Once) ***27***
1.7. Công cụ sử dụng ***29***
1.7.1. Ngôn ngữ lập trình Python *29*
1.7.2. Roboflow *30*
1.7.3. Google Colab *30*
1.7.4. Ultralytics *31*
1.8. Các phương pháp đánh giá mô hình thị giác máy tính ***31***
1.8.1. Ma trận nhầm lẫn *31*
1.8.2. Precision và Recall *32*
1.8.3. F1 Score và Accuracy *32*
1.8.4. IoU *33*
1.8.5. AP và mAP *33*
1.8.6. Đường cong AUC-ROC *34*
Chương 2. Xây dựng mô hình thực nghiệm **36**
2.1. Mô tả dữ liệu thực nghiệm ***36***
2.1.1. Bối cảnh và quy trình thu thập dữ liệu *36*
2.1.2. Đặc điểm kỹ thuật của hệ thống ghi nhận ảnh *37*
2.1.3. Độ khó của dữ liệu và thách thức đối với mô hình học sâu *37*
2.2. Thực nghiệm với mô hình Autoencoder ***38***
2.2.1. Chuẩn bị và xử lý dữ liệu *38*
2.2.2. Huấn luyện mô hình *39*
2.2.3. Kết quả *42*
2.3. Thực nghiệm với mô hình EfficientNet ***44***
2.3.1. Chuẩn bị và xử lý dữ liệu *45*
2.3.2. Huấn luyện mô hình *46*

2.3.3. Kết quả *47*
2.4. Thực nghiệm với mô hình YOLO ***49***
2.4.1. Chuẩn bị và xử lý dữ liệu *49*
2.4.2. Huấn luyện mô hình *51*
2.4.3. Kết quả *51*
2.5. Kiểm thử và đánh giá mô hình ***54***
2.5.1. Tập dữ liệu kiểm thử *54*
2.5.2. Phương pháp kiểm thử *55*
2.5.3. Kết quả kiểm thử *55*
PHẦN V. KẾT LUẬN VÀ KIẾN NGHỊ **57**
TÀI LIỆU THAM KHẢO **60**

DANH MỤC CÁC CHỮ VIẾT TẮT VÀ KÝ HIỆU

AI	Artificial Intelligence
CNN	Convolutional Neural Network
AE	Autoencoder
GAN	Generative Adversarial Networks
GPU	Graphics Processing Unit
IoU	Intersection over Union
mAP	Mean Average Precision
YOLO	You Only Look Once
ROC	Receiver Operating Characteristic
NMS	Non-maximum Suppression
AOU	Area Under the Curve
J	Chỉ số Youden’s J
TPR	True Positive Rate
FPR	False Positive Rate
DL	Deep Learning
CV	Computer Vision
AOI	Automated Optical Inspection

DANH MỤC CÁC BẢNG BIỂU

Bảng 2.1. Kết quả mô hình Autoencoder 42
Bảng 2.2. Kết quả huấn luyện của mô hình Efficientnet 47
Bảng 2.3. Bảng so sánh kết quả huấn luyện hai mô hình YOLO 52
Bảng 2.4. Mô tả tập dữ liệu kiểm thử 54
Bảng 2.5. Bảng kết quả kiểm thử 55

DANH MỤC CÁC SƠ ĐỒ, ĐỒ THỊ, HÌNH VẼ

Hình 1.1. Minh họa về một kiến trúc mạng nơ-ron tích chập đầy đủ. (Nguồn: Github).. 18
Hình 1.2. Minh họa phép tính tích chập trên ảnh.(Nguồn: Researchgate.net) 19
Hình 1.3. Minh họa kết quả của một feature map sau khi qua MaxPooling và AveragePooling. (Nguồn: Hanli Wang) 21
Hình 1.4. Trải dài ma trận bằng lớp Flattening. (Nguồn: codefinity)..... 22
Hình 1.5. Minh họa lớp kết nối đầy đủ. (Nguồn: Qi Xu) 22
Hình 1.6. COCO mAP của từng phiên bản YOLO (Nguồn: ultralytics)..... 28
Hình 1.7. Minh hoạ hệ thống xử lý hình ảnh YOLO (Nguồn: arxiv)..... 29
Hình 1.8. Ma trận nhầm lẫn (Nguồn: bigdatauni) 31
Hình 1.9. Minh họa các tham số cần để tính IoU. (Nguồn:viblo.asia) 33
Hình 2.10. Minh hoạ ảnh lon rỗng thu được từ hệ thống 37
Hình 2.11. Ảnh đầu vào và sau khi được xử lý để đưa vào mô hình Autoencoder..... 39
Hình 2.12. Kiến trúc của bộ mã hoá Encoder..... 40
Hình 2.13. Kiến trúc của bộ giải mã Decoder 41
Hình 2.14. Đồ thị ROC với điểm ngưỡng tối ưu được đánh dấu 42
Hình 2.15. Ma trận hỗn hợp của mô hình Autoencoder 43
Hình 2.16. Minh hoạ kết quả dự đoán mô hình Autoencoder 44
Hình 2.17. Kết quả ma trận nhầm lẫn của mô hình Efficientnet 48
Hình 2.18. Hình minh hoạ kết quả dự đoán mô hình EfficientNet..... 49
Hình 2.19. Minh hoạ quá trình gán nhãn bằng Roboflow 50
Hình 2.20. Ma trận nhầm lẫn của 2 mô hình YOLO 53
Hình 2.21. Hình minh hoạ kết quả dự đoán mô hình YOLO 54

PHẦN I. PHẦN MỞ ĐẦU

1. Lý do chọn đề tài

Trong bối cảnh Cách mạng Công nghiệp 4.0, việc ứng dụng các công nghệ mới như thị giác máy tính (Computer Vision) và học sâu (Deep Learning) đang trở thành xu hướng tất yếu trong hoạt động kiểm tra chất lượng sản phẩm (Quality Control). Các nghiên cứu trên thế giới chỉ ra rằng hệ thống kiểm tra tự động dựa trên hình ảnh có thể đạt độ chính xác cao hơn con người, đồng thời giảm đáng kể chi phí vận hành và hạn chế rủi ro do yếu tố chủ quan của nhân công [1]. Theo nghiên cứu của Kang và cộng sự (2019), việc ứng dụng deep learning vào kiểm tra bề mặt kim loại giúp tăng độ chính xác lên đến 96% so với phương pháp truyền thống, đồng thời giảm thời gian kiểm tra xuống còn 1/10 so với kiểm tra thủ công [2]. Điều này cho thấy tiềm năng rất lớn của các công nghệ học sâu trong phát hiện khuyết tật quy mô công nghiệp.

Trong ngành công nghiệp sản xuất bia, mặc dù quy trình chiết rót – đóng nắp – thanh trùng được tự động hóa ở mức cao, nhưng công đoạn kiểm tra lon rỗng (empty can inspection) vẫn đóng vai trò đặc biệt quan trọng vì liên quan trực tiếp đến chất lượng đầu vào của toàn bộ dây chuyền. Một lon rỗng bị lỗi – dù chỉ là móp nhẹ, mép lon biến dạng, hay vết lõm rất nhỏ – đều có thể gây ra nhiều rủi ro như kẹt cơ cấu cấp lon, hư hại máy chiết, đổ tràn sản phẩm hoặc làm giảm chất lượng lon thành phẩm. De Silva (2020) chỉ ra rằng lỗi cơ học dù nhỏ ở vỏ lon nhôm có thể gây ảnh hưởng đến tốc độ vận hành và giảm độ ổn định của dây chuyền chiết rót [3].

Tuy nhiên, trong thực tế sản xuất tại Carlsberg Việt Nam – nơi tác giả thực tập và thu thập dữ liệu cho đề tài này – số lượng lon lỗi (NG) vô cùng thấp. Điều này là do lon rỗng được nhà cung cấp kiểm định nghiêm ngặt trước khi đưa vào nhà máy. Do đó, phần lớn lon đầu vào đều đạt chuẩn (OK), khiến dữ liệu NG trở nên hiếm và khó thu thập. Đây là vấn đề thường gặp trong các bài toán công nghiệp thực tế, và nhiều nghiên cứu quốc tế

cũng khẳng định rằng rare defect detection là một trong những thách thức lớn của thị giác máy tính hiện đại [4],[5].

Ngoài ra, nhiều loại lỗi trong dữ liệu thực tế lại có hình thái rất giống với lon đạt chuẩn. Sự khác biệt đôi khi chỉ đến từ phản xạ ánh sáng bất thường do biến dạng rất nhỏ, làm tăng thêm độ khó của bài toán. Điều này khiến nhiều mô hình phân loại truyền thống khó đạt kết quả mong đợi. Nghiên cứu của Bergmann (2019) về anomaly detection trong công nghiệp cũng nêu rõ: *"Trong các bài toán mà lỗi rất nhỏ hoặc ít xuất hiện, các mô hình phân loại thông thường dễ bị overfit vào lớp OK và bỏ sót NG"* [6].

Một vấn đề khác cũng được ghi nhận trong thực tế là các nhà máy thường sử dụng hệ thống kiểm tra lon từ các nhà cung cấp thiết bị nước ngoài. Các hệ thống này hoạt động như hộp đen (black-box), doanh nghiệp không nắm được thuật toán bên trong và không có khả năng tự điều chỉnh. Khi xảy ra sự cố, công ty phải mời chuyên gia từ nhà cung cấp sang hỗ trợ, gây tốn kém chi phí, kéo dài thời gian xử lý, thậm chí ảnh hưởng kế hoạch sản xuất. Chính vì vậy, việc tự nghiên cứu, khảo sát và đánh giá các mô hình nhận diện hiện đại để tìm ra giải pháp phù hợp là nhu cầu cấp thiết.

Xuất phát từ những lý do đó, kết hợp với điều kiện thực tiễn thu thập dữ liệu tại Carlsberg Việt Nam, tác giả lựa chọn đề tài **"Ứng dụng thị giác máy tính và học sâu vào kiểm tra chất lượng lon rỗng trong dây chuyền công nghiệp"**. Đề tài không nhằm triển khai thực tế hệ thống hoàn chỉnh, mà tập trung đánh giá – so sánh – phân tích hiệu quả của các phương pháp hiện đại, từ đó đề xuất giải pháp phù hợp nhất cho việc phát hiện lỗi hiếm và khó nhận biết trong môi trường công nghiệp.

2. Mục tiêu nghiên cứu

Đề tài có mục tiêu chính là nghiên cứu và đánh giá khả năng ứng dụng của bốn phương pháp học sâu trong nhận diện lon rỗng thuộc hai lớp OK và NG. Đây là nghiên cứu mang tính *tiền khả thi* (feasibility study) nhằm phục vụ định hướng triển khai sau

này. Tập trung vào **nghiên cứu – đánh giá – so sánh**, không nhằm triển khai hệ thống hoàn chỉnh. Mục tiêu cụ thể bao gồm:

1. Xây dựng bộ dữ liệu hình ảnh lon rỗng từ dây chuyền Carlsberg Việt Nam, bao gồm lon OK và lon NG (với số lượng NG giới hạn).
2. Khảo sát và triển khai bốn mô hình học sâu đại diện cho bốn hướng tiếp cận khác nhau:
    - Autoencoder (Anomaly Detection – phát hiện bất thường khi dữ liệu NG rất ít)
    - EfficientNet (Image Classification – phân loại hình ảnh đầu cuối)
    - YOLOv8-seg (Object Detection + Segmentation)
    - YOLOv11-seg (phiên bản mới nhất, tối ưu cho tốc độ và độ chính xác)
3. Đánh giá và so sánh khả năng nhận diện lỗi của từng mô hình.
4. Xác định **giải pháp phù hợp nhất** cho bài toán, làm nền tảng cho việc triển khai thực tế trong tương lai.

3. Bố cục đề tài

Đề tài được bố cục thành ba phần chính với nội dung như sau:

Phần I – Phần mở đầu trình bày các nội dung định hướng cho toàn bộ đề tài, bao gồm: lý do chọn đề tài, mục tiêu nghiên cứu, bố cục tổng thể của luận văn. Phần này đóng vai trò giới thiệu bối cảnh, xác định vấn đề và đặt nền tảng cho các phần tiếp theo.

Phần II – Tổng quan vấn đề nghiên cứu là phần tổng hợp các nghiên cứu liên quan đến kiểm tra chất lượng sản phẩm bằng học sâu, đặc biệt trong lĩnh vực phát hiện lỗi công nghiệp. Phân tích các hướng tiếp cận phổ biến như classification, segmentation và anomaly detection, từ đó làm rõ cơ sở lựa chọn mô hình cho đề tài.

Phần III – Đối tượng, phạm vi và phương pháp nghiên cứu. Làm rõ đối tượng nghiên cứu là hình ảnh lon rỗng trong dây chuyền sản xuất, phạm vi nghiên cứu tập trung vào xử lý dữ liệu ảnh và mô hình học sâu, không bao gồm phần cứng hay tích hợp hệ

thống. Trình bày phương pháp nghiên cứu, bao gồm quy trình thực nghiệm, lựa chọn mô hình và tiêu chí đánh giá.

Phần IV – Nội dung và kết quả nghiên cứu là phần trọng tâm của đề tài, gồm ba chương chính:

- **Chương 1 – Tổng quan** trình bày cơ sở lý thuyết liên quan đến học sâu và xử lý ảnh, đồng thời xây dựng khung nghiên cứu cho bài toán nhận diện lon rỗng.
- **Chương 2 – Nội dung thực hiện** mô tả chi tiết quy trình thực nghiệm, bao gồm thu thập dữ liệu, tiền xử lý ảnh, xây dựng và huấn luyện bốn mô hình học sâu (Autoencoder, EfficientNet, YOLOv8-seg, YOLOv11-seg), kèm theo việc đánh giá và phân tích hiệu quả của từng mô hình. **Nêu quy trình thử nghiệm các mô hình với tập dữ liệu mới để đưa ra kết quả** tổng hợp, đánh giá ưu – nhược điểm của từng phương pháp, đồng thời nêu ra những hạn chế còn tồn tại trong quá trình thực nghiệm và phân tích tiềm năng cải thiện.

Phần V – Kết luận và kiến nghị đưa ra kết luận chung của đề tài dựa trên toàn bộ quá trình nghiên cứu, đồng thời đề xuất các kiến nghị và hướng phát triển phù hợp cho các nghiên cứu và ứng dụng thực tế trong tương lai.

PHẦN II. TỔNG QUAN VẤN ĐỀ NGHIÊN CỨU

Trong những năm gần đây, sự phát triển mạnh mẽ của học sâu (deep learning) đã mở ra nhiều cơ hội mới cho việc kiểm tra chất lượng sản phẩm trong công nghiệp. Thay vì dựa vào các phương pháp truyền thống vốn hạn chế về độ chính xác và khả năng tổng quát, các mô hình học sâu – đặc biệt là mạng nơ-ron tích chập (Convolutional Neural Networks, CNN) đã chứng minh hiệu quả vượt trội. Từ đây, hàng loạt công trình nghiên cứu tiêu biểu ra đời, đặt nền móng cho các phương pháp kiểm tra tự động hiện đại.

Nghiên cứu của S. Ren và các cộng sự (2015) về Faster R-CNN đã đặt nền móng cho các phương pháp phát hiện đối tượng hiện đại, tạo tiền đề cho dòng YOLO sau này [7]. Đây là một trong những mô hình đầu tiên cho phép phát hiện đối tượng với tốc độ nhanh hơn đáng kể nhưng vẫn đảm bảo độ chính xác cao, mở đường cho việc ứng dụng vào các hệ thống kiểm tra thời gian thực trong công nghiệp. Phương pháp YOLO (You Only Look Once) do Redmon và cộng sự (2016) giới thiệu đã tạo bước ngoặt khi đưa tốc độ nhận diện lên mức real-time với độ chính xác cạnh tranh [9]. Từ đó đến nay, YOLO phát triển liên tục qua nhiều phiên bản như YOLOv3, YOLOv5, YOLOv7, YOLOv8 và gần đây nhất là YOLOv11, mỗi phiên bản đều cải thiện đáng kể cả về cấu trúc mạng lưới, khả năng tổng quát hóa và ưu tiên tối ưu hoá tốc độ – yếu tố rất quan trọng đối với dây chuyền sản xuất tự động.

Trong lĩnh vực phát hiện lỗi (defect detection), nhiều nghiên cứu đã chứng minh rằng segmentation-based detection (như Mask R-CNN, U-Net, YOLO-Seg) hiệu quả hơn so với classification-based detection khi xử lý các lỗi nhỏ hoặc có hình thái phức tạp. Ứng dụng thị giác máy tính và học sâu vào kiểm tra chất lượng lon rỗng trong dây chuyền công nghiệp (2020) đã chỉ ra rằng segmentation cho phép mô hình định vị chính xác vùng lỗi ngay cả khi lỗi rất mờ hoặc bị nhiễu ánh sáng che khuất [10].

Đối với các bài toán có tỷ lệ mẫu lỗi rất thấp (rare defects), hướng tiếp cận anomaly detection trở nên phổ biến. Công trình ITAD (Industrial Texture Anomaly

Detection) của Bergmann et al. (2019) đã đề xuất sử dụng Autoencoder và các mô hình reconstruction để phát hiện bất thường trong sản phẩm công nghiệp [11]. Các tác giả chỉ ra rằng trong môi trường công nghiệp, dữ liệu lỗi rất khó thu thập vì tần suất xuất hiện thấp, do đó các phương pháp như Autoencoder hoặc Variational Autoencoder có khả năng học phân bố của mẫu tốt và phát hiện điểm bất thường dựa trên reconstruction error.

Những nghiên cứu này có giá trị trực tiếp trong bối cảnh bài toán nhận diện lon rỗng, bởi dữ liệu NG của lon cũng là dữ liệu lỗi hiếm (rare defects), ngoại hình lỗi tinh vi và thường rất khó phát hiện bằng mắt thường.

Riêng trong ngành thực phẩm – đồ uống, một số nghiên cứu đã ứng dụng học sâu để kiểm tra bao bì, dị vật hoặc lỗi sản phẩm. Ví dụ, nghiên cứu của nhóm Zhang (2022) đã áp dụng YOLOv5 để phát hiện lỗi trên chai nhựa trong dây chuyền sản xuất, đạt độ chính xác trên 95% [12]. Điều này cho thấy sự phù hợp của các mô hình YOLO cho các bài toán kiểm tra vật thể có hình dạng thay đổi nhỏ nhưng yêu cầu độ chính xác cao.

Tổng hợp lại, các nghiên cứu cho thấy ba xu hướng chính:

(1) **Phân loại hình ảnh (Classification)** phù hợp cho các lỗi rõ rệt.

(2) **Segmentation-based detection** phù hợp cho lỗi nhỏ và khó nhận biết.

(3) **Anomaly detection (Autoencoder)** phù hợp khi dữ liệu lỗi hiếm.

Ba xu hướng này chính là ba hướng mà đề tài đã triển khai bằng bốn mô hình AE, EfficientNet, YOLOv8-seg và YOLOv11-seg.

PHẦN III. ĐỐI TƯỢNG, PHẠM VI VÀ PHƯƠNG PHÁP NGHIÊN CỨU

1. Đối tượng và phạm vi nghiên cứu

- **Đối tượng nghiên cứu:**

Đối tượng nghiên cứu của đề tài là **hình ảnh của lon bia rỗng** được thu thập trực tiếp từ dây chuyền sản xuất tại Carlsberg Việt Nam. Đây là dữ liệu phản ánh tình trạng thực tế của lon trước khi đi vào các công đoạn tiếp theo như chiết rót, đóng nắp và kiểm tra áp lực. Các hình ảnh này bao gồm cả những lon đạt tiêu chuẩn (OK) và những lon không đạt (NG), mặc dù số lượng OK chiếm ưu thế tuyệt đối do lon rỗng vốn được nhà cung cấp kiểm định chất lượng nghiêm ngặt trước khi vận chuyển đến nhà máy.

Bài toán đặt ra là xác định **liệu một lon rỗng có đạt tiêu chuẩn hay không** dựa trên dữ liệu hình ảnh. Điều này đòi hỏi mô hình phải có khả năng phân biệt được các đặc điểm tinh vi của lon đạt và lon lỗi. Đặc biệt, các lỗi này thường rất nhỏ, có hình thái không nhất quán và trong nhiều trường hợp, sự khác biệt giữa lon OK và NG là rất khó nhận biết bằng mắt thường. Điều này làm tăng độ khó của bài toán và cũng là lý do quan trọng để nghiên cứu nhiều phương pháp học sâu khác nhau trong đề tài này.

- **Phạm vi nghiên cứu:**

Phạm vi nghiên cứu của đề tài được xác định rõ ràng để đảm bảo tập trung vào bản chất kỹ thuật của bài toán, cụ thể như sau:

Thứ nhất, đề tài chỉ tập trung vào việc xử lý dữ liệu hình ảnh nhằm phát hiện các lon rỗng không đạt tiêu chuẩn. Các yếu tố liên quan đến phần cứng như hệ thống camera, tốc độ chụp, bố trí ánh sáng hay cơ cấu cơ khí của dây chuyền sản xuất không nằm trong phạm vi nghiên cứu. Sự lựa chọn này nhằm đảm bảo đề tài có thể đi sâu vào phương diện thuật toán và mô hình học sâu mà không bị ảnh hưởng bởi các biến số của hệ thống công nghiệp thực tế.

Thứ hai, nghiên cứu chỉ triển khai bốn phương pháp học sâu gồm Autoencoder, EfficientNet, YOLOv8-seg và YOLOv11-seg. Đây là bốn đại diện tiêu biểu cho ba hướng tiếp cận khác nhau: phát hiện bất thường (anomaly detection), phân loại (classification), phát hiện đối tượng (object detection) và phân đoạn (segmentation). Mỗi phương pháp có ưu nhược điểm riêng và được lựa chọn nhằm mục tiêu so sánh toàn diện, từ đó tìm ra giải pháp phù hợp nhất cho bài toán cụ thể.

Thứ ba, đề tài chỉ nhận diện lon lỗi và lon không lỗi, không đi sâu vào việc phân loại chi tiết mức độ lỗi của lon. Đồng thời, nghiên cứu không triển khai mô hình vào dây chuyền sản xuất thực tế và cũng không hướng đến việc xây dựng phần mềm giám sát hay giao diện người dùng. Các kết quả chỉ dừng lại ở mức thực nghiệm với dữ liệu thu thập sẵn, được trình bày dưới dạng mô hình thử nghiệm, đánh giá hiệu năng và phân tích kết quả. Điều này phù hợp với điều kiện thực tế vì việc tích hợp mô hình vào hệ thống công nghiệp đòi hỏi nhiều yếu tố khác như đồng bộ hóa phần cứng, xử lý real-time, an toàn vận hành và kiểm thử kỹ thuật - những yếu tố vượt ra ngoài phạm vi của một luận văn tốt nghiệp.

2. Phương pháp nghiên cứu

- **Phương pháp nghiên cứu lý thuyết:**

Phương pháp nghiên cứu lý thuyết được sử dụng để tìm hiểu và hệ thống hóa các kiến thức liên quan đến học sâu, mạng nơ-ron tích chập và thị giác máy tính. Trong quá trình này, tác giả tham khảo các bài báo, công trình khoa học và tài liệu chuyên ngành về các phương pháp phát hiện bất thường, phân loại hình ảnh và phân đoạn đối tượng, đặc biệt là các mô hình Autoencoder, EfficientNet, YOLOv8-seg và YOLOv11-seg. Việc nghiên cứu lý thuyết đóng vai trò xây dựng nền tảng khoa học để lý giải cách thức hoạt động của các mô hình, cơ sở lựa chọn thuật toán, cũng như tiền đề cho quá trình triển khai thực nghiệm.

- **Phương pháp đánh giá hiệu quả:**

Phương pháp đánh giá hiệu quả được sử dụng nhằm đo lường và so sánh chính xác khả năng nhận diện lon đạt chuẩn OK và lon không đạt chuẩn NG của từng mô hình. Tác giả áp dụng các chỉ số đánh giá phổ biến trong học sâu như Accuracy, Precision, Recall và F1-score đối với các mô hình phân loại; đồng thời sử dụng mAP và IoU cho các mô hình phát hiện và phân đoạn. Các chỉ số này được tính toán dựa trên tập dữ liệu kiểm thử, kết hợp phân tích ma trận nhầm lẫn nhằm đánh giá chi tiết những trường hợp mô hình dự đoán sai, từ đó có cơ sở khách quan để so sánh ưu điểm và hạn chế của từng phương pháp.

- **Phương pháp nghiên cứu thực nghiệm:**

Phương pháp thực nghiệm được tiến hành thông qua quá trình xây dựng và xử lý bộ dữ liệu hình ảnh lon rỗng thu thập từ dây chuyền tại Carlsberg Việt Nam. Dữ liệu được chuẩn hóa, gán nhãn (đối với các mô hình YOLO) và chia thành các tập huấn luyện–kiểm thử theo tỷ lệ phù hợp. Trên cơ sở đó, từng mô hình học sâu được triển khai: Autoencoder dùng để học đặc trưng của lon OK nhằm phát hiện bất thường; EfficientNet dùng để phân loại OK/NG; YOLOv8-seg và YOLOv11-seg dùng để xác định vùng lỗi. Sau khi huấn luyện, các mô hình được kiểm thử để thu thập kết quả đánh giá theo cùng một quy trình nhằm đảm bảo tính khách quan và khả năng so sánh.

- **Phương pháp phân tích dữ liệu:**

Phương pháp phân tích dữ liệu được sử dụng để tổng hợp, diễn giải và so sánh các kết quả thu được từ thực nghiệm. Tác giả phân tích các chỉ số của từng mô hình, đối chiếu hiệu quả giữa các phương pháp, xác định nguyên nhân dẫn đến sai lệch trong dự đoán, cũng như đánh giá khả năng mô hình hóa các lỗi nhỏ và hiếm gặp. Trên cơ sở đó, việc phân tích giúp rút ra được mô hình phù hợp nhất với bài toán nhận diện lon rỗng và đề xuất hướng cải tiến cho các nghiên cứu tiếp theo.

PHẦN IV. NỘI DUNG VÀ KẾT QUẢ NGHIÊN CỨU

CHƯƠNG 1. TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT

1.1. Tổng quan về Thị giác máy tính trong Công nghiệp

1.1.1. Vai trò của xử lý ảnh trong kiểm soát chất lượng (Quality Control)

Thị giác máy tính (Computer Vision – CV) là lĩnh vực nghiên cứu các thuật toán cho phép máy tính "nhìn", hiểu và diễn giải thông tin từ dữ liệu hình ảnh, từ đó hỗ trợ tự động hóa trong nhiều ngành công nghiệp. Trong bối cảnh chuyển đổi số công nghiệp (Industry 4.0), CV đóng vai trò trọng yếu trong việc xây dựng các hệ thống kiểm soát chất lượng tự động (Automated Optical Inspection – AOI).

Trước đây, kiểm tra chất lượng sản phẩm trong dây chuyền chủ yếu dựa vào quan sát thủ công, phụ thuộc nhiều vào năng lực và sự tập trung của công nhân, dẫn đến biến động lớn về độ chính xác và chi phí lao động. Các nghiên cứu cho thấy khả năng duy trì sự chú ý của con người giảm mạnh sau 20–30 phút quan sát liên tục, khiến tỉ lệ phát hiện lỗi chỉ đạt 70–85% trong môi trường sản xuất có nhịp độ cao. [13]

Sự phát triển của học sâu (Deep Learning) đã tạo ra bước ngoặt lớn. Các hệ thống AOI hiện đại có thể đạt độ chính xác vượt trội 95–99%, hoạt động liên tục 24/7 và đảm bảo tính nhất quán trong đánh giá sản phẩm [14]. Trong các dây chuyền sản xuất đồ uống, bao bì kim loại và đặc biệt là ngành bia – nơi mỗi phút có thể xử lý hàng nghìn lon – việc ứng dụng thị giác máy tính giúp phát hiện sớm các lỗi hình học của lon, tránh kẹt cơ khí và giảm thiểu thiệt hại sản xuất.

Trong bối cảnh đó, đề tài tập trung vào nhận diện lon rỗng OK/NG dựa trên hình ảnh, sử dụng các kỹ thuật CV và DL hiện đại nhằm giải quyết thách thức lớn nhất của bài toán: **lỗi hiếm**, **lỗi nhỏ**, và **dữ liệu không đồng đều**.

1.1.2. Các bài toán cơ bản trong kiểm tra bề mặt sản phẩm

Trong nghiên cứu này, bài toán kiểm tra chất lượng vỏ lon được tiếp cận dưới ba góc độ khác nhau của thị giác máy tính, tương ứng với ba phương pháp được triển khai:

a. Phân loại ảnh (Image Classification):

Đây là bài toán cơ bản nhất, trong đó mô hình nhận đầu vào là ảnh của một sản phẩm (vỏ lon) và gán cho nó một nhãn duy nhất: "Đạt" (OK) hoặc "Không đạt" (NG/Not Good).

Đặc điểm: Mô hình nhìn tổng thể bức ảnh và đưa ra quyết định dựa trên các đặc trưng toàn cục.

Ứng dụng trong đồ án: Sử dụng mạng **EfficientNet** để phân loại nhanh chóng các lon thành hai nhóm OK/NG. Phương pháp này phù hợp cho các cơ chế loại bỏ (reject) nhanh trên băng chuyền, nơi chỉ cần biết sản phẩm có lỗi hay không mà chưa cần biết chi tiết lỗi nằm ở đâu.

b. Phát hiện bất thường (Anomaly Detection):

Khác với phân loại (vốn cần biết trước các loại lỗi để huấn luyện), phát hiện bất thường tập trung vào việc học các đặc trưng của sản phẩm "bình thường". Bất kỳ mẫu nào có độ sai lệch lớn so với phân phối chuẩn đã học sẽ được coi là bất thường.

Đặc điểm: Thường sử dụng phương pháp học không giám sát (Unsupervised Learning) hoặc bán giám sát. Rất hữu ích trong môi trường công nghiệp thực tế, nơi dữ liệu sản phẩm lỗi thường khan hiếm và khó thu thập đầy đủ các biến thể.

Ứng dụng trong đồ án: Sử dụng kiến trúc **Autoencoder** để học cách tái tạo lại vỏ lon chuẩn. Khi gặp vỏ lon lỗi, mô hình sẽ không tái tạo được các vết lỗi đó, dẫn đến "sai số tái tạo" cao, từ đó phát hiện ra sản phẩm hỏng.

c. Phân đoạn đối tượng (Instance Segmentation):

Đây là bài toán phức tạp và chi tiết nhất. Thay vì chỉ đưa ra một nhãn cho cả bức ảnh, mô hình sẽ xác định vị trí của từng vết lỗi và tô màu (phân đoạn) chính xác từng pixel thuộc về vết lỗi đó.

Đặc điểm: Cung cấp thông tin chi tiết về vị trí, hình dáng và kích thước của lỗi. Điều này không chỉ giúp loại bỏ sản phẩm hỏng mà còn cung cấp dữ liệu để phân tích nguyên nhân gốc rễ (ví dụ: vết móp ở vị trí cụ thể có thể do va đập tại một khúc cua băng tải nào đó).

Ứng dụng trong đồ án: Sử dụng các kiến trúc **YOLOv8** và **YOLOv11** (phiên bản Segmentation). Đây là các mô hình tiên tiến nhất hiện nay, cân bằng giữa tốc độ xử lý thời gian thực (Real-time) và độ chính xác trong việc định vị lỗi.

1.2. Tổng quan về học máy

1.2.1. Thế nào là học máy

Học máy (Machine Learning - ML) là một nhánh quan trọng của trí tuệ nhân tạo (Artificial Intelligence - AI), tập trung vào việc thiết kế các thuật toán và mô hình giúp máy tính có thể học từ dữ liệu mà không cần lập trình cụ thể cho từng tác vụ. Thay vì tuân theo các quy tắc cứng nhắc, học máy cho phép hệ thống tự phát hiện các mẫu, quy luật trong dữ liệu và sử dụng chúng để đưa ra dự đoán hoặc quyết định. Điều này tạo ra các hệ thống thông minh có khả năng cải thiện hiệu suất theo thời gian dựa trên kinh nghiệm và dữ liệu thu thập được.

Theo Tom M. Mitchell, học máy được định nghĩa "Một chương trình máy tính được cho là học hỏi từ kinh nghiệm E có liên quan tới một vài nhiệm vụ T và hiệu suất đo lường P, nếu hiệu suất của nó trên T được đo bằng P cải thiện sau khi trải qua kinh nghiệm E" [15] hay trong cuốn *"Introduction to Machine Learning"*, Ethem Alpaydin định nghĩa: *"Học máy là ngành khoa học nghiên cứu các thuật toán và hệ thống mà tự động cải thiện thông qua dữ liệu"* [16].

Học máy hoạt động dựa trên quá trình phân tích dữ liệu, từ đó tìm ra mối quan hệ giữa các đầu vào và kết quả mong muốn. Một mô hình học máy trải qua ba giai đoạn cơ bản:

- Huấn luyện (Training): Mô hình được cung cấp dữ liệu đầu vào cùng nhãn hoặc thông tin tương ứng, từ đó học cách tối ưu hóa các tham số để phản ánh mối quan hệ giữa đầu vào và đầu ra.
- Xác thực (Validation): Dữ liệu kiểm tra được sử dụng để đánh giá mô hình trong quá trình huấn luyện, giúp điều chỉnh các siêu tham số và tránh hiện tượng quá khớp.
- Dự đoán (Inference): Sau khi được huấn luyện, mô hình áp dụng kiến thức đã học để đưa ra dự đoán trên dữ liệu mới.

1.2.2. Phân loại học máy

Học máy là một thuật ngữ bao quát, được chia thành nhiều nhánh nhỏ, mỗi nhánh tập trung nghiên cứu một lĩnh vực cụ thể. Tùy thuộc vào từng loại bài toán và yêu cầu thực tế, các phương pháp trong học máy cũng được phát triển theo những cách tiếp cận khác nhau để giải quyết hiệu quả các vấn đề đa dạng.

1.2.2.1. Học có giám sát

Học có giám sát (Supervised Learning) là một nhóm các thuật toán máy học sử dụng dữ liệu đã được gán nhãn để mô hình hóa mối quan hệ giữa biến đầu vào (x) và biến đầu ra (y). Hai nhóm bài toán chính trong học có giám sát là phân loại (classification) và hồi quy (regression) [17]. Trong phân loại, biến đầu ra thường mang giá trị rời rạc, thuộc một trong nhiều lớp, như "nam" hoặc "nữ," "spam" hoặc "không spam." Ngược lại, bài toán hồi quy tập trung dự đoán các giá trị liên tục ví dụ như giá nhà dựa trên diện tích và vị trí.

Trong học có giám sát, hệ thống được huấn luyện từ các mẫu dữ liệu đã gán nhãn, nghĩa là đầu ra của từng mẫu đã được biết trước. Chất lượng và sự đầy đủ của dữ liệu gán nhãn là yếu tố then chốt, quyết định hiệu suất của mô hình trong việc giải quyết các bài

toán thực tế. Ví dụ, trong phân loại thư rác, hệ thống được huấn luyện trên các email đã gán nhãn trước đó với thông tin như tiêu đề, nội dung, và từ khóa. Sau khi huấn luyện, mô hình có thể tự động phân loại các email mới dựa trên các đặc điểm học được [18],[19]. Một ứng dụng khác là nhận dạng hình ảnh, nơi mô hình học từ các hình ảnh đã gán nhãn (ví dụ, hình ảnh của các phương tiện giao thông). Hệ thống có thể trích xuất các đặc trưng quan trọng như hình dạng, màu sắc, hoặc kích thước từ dữ liệu huấn luyện, và sử dụng các đặc trưng này để phân loại các hình ảnh chưa được gán nhãn.

Học có giám sát đóng vai trò quan trọng trong việc xây dựng các mô hình dự đoán chính xác cho nhiều ứng dụng thực tiễn, từ phân loại văn bản, nhận dạng hình ảnh đến dự đoán tài chính. Tuy nhiên, để đảm bảo hiệu quả, việc thu thập và gán nhãn dữ liệu cần được thực hiện một cách kỹ lưỡng và chính xác.

1.2.2.2. Học không giám sát

Trong học không giám sát, hệ thống học máy thường tối ưu hóa một hàm mục tiêu hoặc tiêu chí để tìm ra các cấu trúc tiềm ẩn trong dữ liệu. Ví dụ, trong bài toán phân cụm (clustering), thuật toán cố gắng giảm thiểu khoảng cách giữa các điểm dữ liệu trong cùng một nhóm, đồng thời tăng khoảng cách giữa các nhóm khác nhau [20]. Trong các bài toán giảm số chiều (dimensionality reduction), thuật toán tìm cách nén dữ liệu xuống không gian có số chiều thấp hơn trong khi vẫn bảo toàn được thông tin quan trọng [21].

Học không giám sát có nhiều ứng dụng quan trọng trong các lĩnh vực khác nhau nhờ khả năng khai thác các cấu trúc ẩn và mẫu dữ liệu mà không cần nhãn. Trong phân tích thị trường, học không giám sát giúp các doanh nghiệp phân nhóm khách hàng theo sở thích và hành vi tiêu dùng, từ đó tối ưu hóa chiến lược tiếp thị và nâng cao trải nghiệm khách hàng. Trong y sinh, phương pháp này được sử dụng để phân tích và tìm ra các mẫu trong dữ liệu gene hoặc xác định các bệnh lý từ dữ liệu y tế mà không cần nhãn bệnh cụ thể.

Ngoài ra, học không giám sát còn ứng dụng mạnh mẽ trong hệ thống khuyến nghị, ví dụ như gợi ý sản phẩm, phim ảnh hoặc bài hát cho người dùng dựa trên hành vi trước đó. Xử lý ngôn ngữ tự nhiên (NLP) cũng là một lĩnh vực ứng dụng học không giám sát, giúp cải thiện các mô hình dự đoán, dịch ngôn ngữ, hoặc phân tích văn bản mà không cần dữ liệu nhãn rõ ràng. Giám sát an ninh là một ví dụ khác, trong đó các thuật toán học không giám sát giúp phát hiện hành vi bất thường hoặc xâm nhập trong các hệ thống giám sát video.

Nhìn chung, học không giám sát đóng vai trò quan trọng trong việc khám phá và hiểu sâu hơn về các cấu trúc tiềm ẩn trong dữ liệu lớn, góp phần nâng cao hiệu quả và khả năng ứng dụng trong nhiều ngành công nghiệp.

1.2.2.3. Học bán giám sát

Học bán giám sát là một phương pháp trong học máy, kết hợp giữa học giám sát và học không giám sát, nhằm tận dụng ưu điểm của cả hai cách tiếp cận. Trong khi học giám sát yêu cầu dữ liệu có nhãn đầy đủ để huấn luyện, và học không giám sát chỉ sử dụng dữ liệu không nhãn, học bán giám sát khai thác hiệu quả cả dữ liệu có nhãn và không nhãn, giúp xây dựng mô hình học với chi phí gán nhãn thấp hơn [22].

Phương pháp này sử dụng một lượng nhỏ dữ liệu có nhãn để định hướng mô hình học, đồng thời tận dụng dữ liệu không nhãn để trích xuất các mẫu và cấu trúc tiềm ẩn. Điều này đặc biệt hữu ích trong các lĩnh vực mà việc gán nhãn dữ liệu tốn kém hoặc mất nhiều thời gian, trong khi dữ liệu không nhãn lại dễ thu thập hơn. Ví dụ, dữ liệu không nhãn có thể giúp cải thiện khả năng phân loại khi được tích hợp với một lượng nhỏ dữ liệu có nhãn [23].

Học bán giám sát có ứng dụng rộng rãi trong nhiều lĩnh vực. Trong xử lý ngôn ngữ tự nhiên, phương pháp này cải thiện mô hình phân loại văn bản từ dữ liệu không nhãn, như tài liệu trực tuyến. Trong nhận dạng hình ảnh, nó tối ưu hóa khả năng phân loại khi chỉ có một phần nhỏ dữ liệu được gán nhãn. Trong y học, học bán giám sát hỗ trợ phát

hiện các mẫu tiềm ẩn từ dữ liệu gene hoặc hình ảnh y tế, nơi việc thu thập nhãn bệnh chính xác rất khó khăn. Đặc biệt, trong tài chính, nó giúp phát hiện gian lận bằng cách khai thác dữ liệu không nhãn từ các giao dịch lớn. Nhìn chung, học bán giám sát giảm chi phí gán nhãn và nâng cao hiệu quả mô hình học máy, mở ra tiềm năng ứng dụng mạnh mẽ trong nhiều ngành.

1.2.3. Học sâu

Học sâu là một nhánh con của học máy, là một kỹ thuật dựa trên các mạng nơ-ron nhân tạo nhiều lớp để mô phỏng cách thức hoạt động của bộ não con người trong việc học hỏi và xử lý thông tin. Học sâu sử dụng các mô hình mạng nơ-ron có nhiều lớp ẩn (hidden layers) để tự động học các đặc trưng (features) và cấu trúc phức tạp trong dữ liệu mà không cần sự can thiệp hay thiết kế đặc trưng thủ công từ con người.

Geoffrey Hinton - Nhà khoa học nổi bật trong lĩnh vực học sâu, được mệnh danh là "cha đẻ của học sâu", định nghĩa học sâu như sau*: "Học sâu là một phương pháp học máy sử dụng các mạng nơ-ron nhiều lớp để tự động trích xuất đặc trưng từ dữ liệu đầu vào và học các mô hình phức tạp từ các mẫu dữ liệu lớn. Nó giúp mô hình nhận diện các mối quan hệ ẩn giữa các đặc trưng của dữ liệu mà không cần sự can thiệp thủ công từ con người"* [24].

Khác với các phương pháp học máy truyền thống, học sâu có khả năng tự động trích xuất các đặc trưng từ dữ liệu đầu vào, chẳng hạn như ảnh, văn bản hoặc âm thanh, mà không cần phải xây dựng một tập hợp các đặc trưng riêng biệt để đưa vào mô hình. Điều này làm cho học sâu đặc biệt hữu ích trong các bài toán có dữ liệu không có cấu trúc, như nhận dạng hình ảnh, nhận dạng giọng nói, dịch ngôn ngữ tự động, và phân tích cảm xúc.

Một trong những yếu tố then chốt của học sâu là các mạng nơ-ron sâu (deep neural networks), trong đó mỗi lớp của mạng thực hiện một phép toán toán học để chuyển đổi dữ liệu đầu vào thành một biểu diễn có giá trị hơn cho các lớp tiếp theo. Các mạng này có thể

có hàng trăm hoặc hàng nghìn lớp ẩn, cho phép chúng học các biểu diễn phức tạp từ dữ liệu đầu vào. Quá trình huấn luyện mạng nơ-ron sâu sử dụng thuật toán lan truyền ngược (backpropagation) để tối ưu hóa trọng số của các kết nối giữa các nơ-ron, dựa trên sai số giữa dự đoán và giá trị thực tế.

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

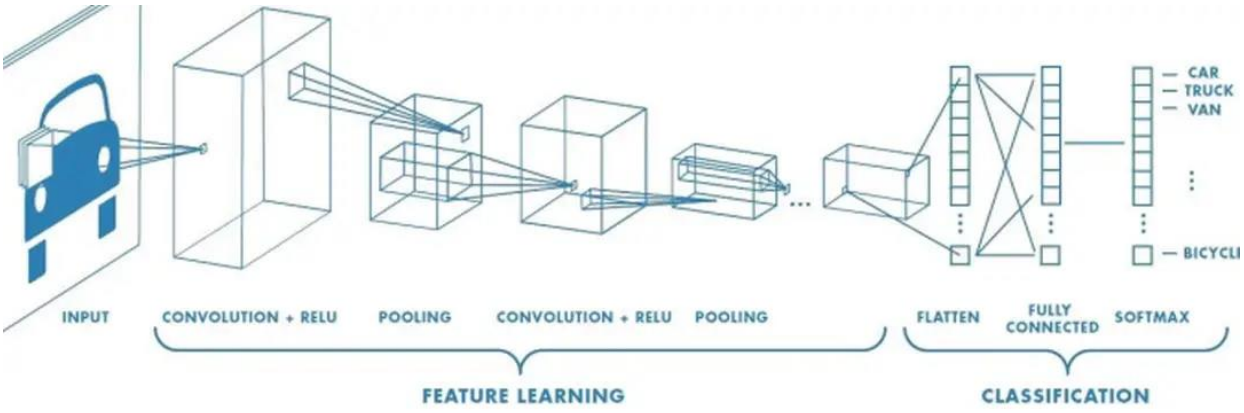
Mạng nơ-ron tích chập (CNN) là một trong những kiến trúc học sâu đột phá nhất, được thiết kế chuyên biệt để xử lý các dữ liệu có cấu trúc lưới như hình ảnh. Khác với các mạng nơ-ron truyền thống (ANN) phải trải phẳng ảnh thành chuỗi vector dài làm mất đi thông tin không gian, CNN duy trì cấu trúc không gian của ảnh thông qua cơ chế tích chập, giúp mô hình trích xuất hiệu quả các đặc trưng từ mức thấp (cạnh, góc) đến mức cao (vật thể hoàn chỉnh).

Mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) là một loại mạng nơ-ron nhân tạo được thiết kế đặc biệt để xử lý dữ liệu có cấu trúc lưới, chẳng hạn như hình ảnh hoặc chuỗi thời gian. CNN hoạt động dựa trên cơ chế tích chập (convolution), một phép toán toán học cho phép mạng học cách trích xuất các đặc trưng không gian từ dữ liệu đầu vào. Ý tưởng chính của CNN là tự động học và trích xuất các đặc trưng quan trọng từ dữ liệu mà không cần sự can thiệp thủ công, giúp giải quyết hiệu quả các bài toán phức tạp như xử lý ảnh, nhận dạng đối tượng và phân loại hình ảnh.

Theo định nghĩa của Yann LeCun, một trong những nhà tiên phong trong lĩnh vực học sâu và là người phát triển mạng nơ-ron tích chập đầu tiên, *"Mạng nơ-ron tích chập là một loại mạng học sâu được thiết kế để tự động học cách trích xuất các đặc trưng không gian từ dữ liệu đầu vào, đặc biệt là hình ảnh. Cấu trúc của CNN lấy cảm hứng từ cách hệ thần kinh thị giác của động vật hoạt động, nơi các nơ-ron đáp ứng với các vùng cục bộ của hình ảnh một cách tuần tự. Điều này cho phép mô hình học cách nhận diện các mẫu cục bộ và mở rộng chúng để nhận diện các đối tượng phức tạp hơn"* [25],[26].

CNN không chỉ bắt nguồn từ ý tưởng mô phỏng hệ thần kinh thị giác sinh học mà còn được thiết kế để tận dụng các đặc trưng cục bộ trong dữ liệu, mang lại hiệu quả vượt trội trong các ứng dụng hiện đại, từ nhận diện khuôn mặt, phân tích ảnh y tế, đến xử lý ngôn ngữ tự nhiên. Mạng CNN ra đời với kiến trúc thay đổi, có khả năng xây dựng liên kết chỉ sử dụng một phần cục bộ trong ảnh kết nối đến node trong lớp tiếp theo thay vì toàn bộ ảnh như trong mạng nơ-ron truyền thẳng.

1.3.1. Kiến trúc cơ bản của mạng nơ-ron tích chập



Hình 1.1. *Minh họa về một kiến trúc mạng nơ-ron tích chập đầy đủ. (Nguồn: Github)*

Một kiến trúc mạng nơ-ron tích chập (CNN) thường bao gồm ba thành phần chính: lớp tích chập (convolutional layer), lớp gộp (pooling layer), và lớp kết nối đầy đủ (fully connected layer). Giữa các lớp tích chập và lớp gộp, các hàm kích hoạt phi tuyến thường được áp dụng để tăng khả năng biểu diễn của mạng. Khi một hình ảnh được đưa vào mạng, nó sẽ trải qua lớp tích chập đầu tiên, nơi các phép toán được thực hiện để trích xuất các đặc trưng từ ảnh. Kết quả của lớp tích chập sẽ được xử lý qua một hàm kích hoạt nhằm làm nổi bật các đặc trưng phi tuyến. Tiếp theo, dữ liệu được đưa đến lớp gộp để giảm kích thước và độ phức tạp của thông tin nhưng vẫn giữ được các yếu tố quan trọng. Cuối cùng, ảnh đã qua xử lý sẽ được chuyển đến lớp kết nối đầy đủ, nơi tất cả các nơ-ron được kết nối với nhau. Dữ liệu tại lớp này thường được đưa qua một hàm kích hoạt Softmax để chuyển đổi thành một vector chứa các xác suất. Đối với bài toán phân loại,

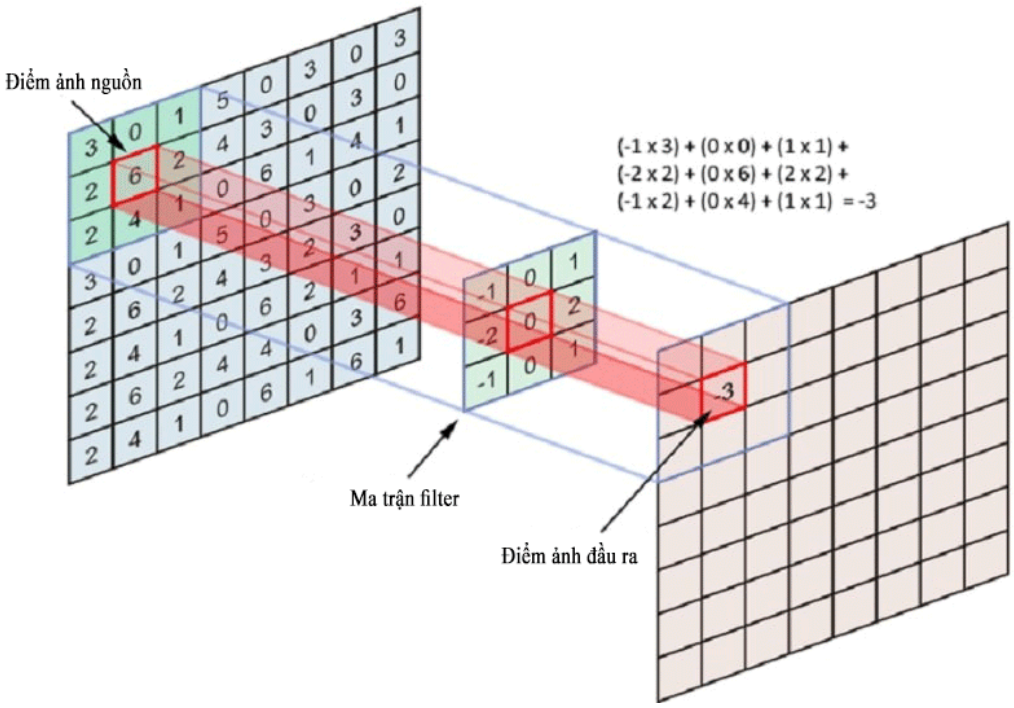
vector đầu ra sẽ biểu diễn xác suất tương ứng với từng lớp, giúp mạng đưa ra dự đoán cuối cùng [27].

a. Lớp Tích chập (Convolutional Layer)

Đây là thành phần cốt lõi của CNN. Lớp này sử dụng tập hợp các bộ lọc (filters/kernels) có kích thước nhỏ trượt qua toàn bộ ảnh đầu vào. Tại mỗi vị trí, bộ lọc thực hiện phép nhân phần tử và tính tổng để tạo ra một giá trị mới trên bản đồ đặc trưng (feature map). Công thức toán học của phép tích chập hai chiều được biểu diễn như sau:

$$S(i,j) = (I \times K)(i,j) = \sum_m \sum_n \mathrm{I}(i-m, j-n) K(m,n) \qquad (1.1)$$

Trong đó: I là ảnh đầu vào, K là bộ lọc (kernel), và S là bản đồ đặc trưng đầu ra. Cơ chế chia sẻ trọng số (weight sharing) của bộ lọc giúp CNN có tính chất bất biến dịch chuyển (translation invariance), nghĩa là có thể nhận diện được vết lỗi (như vết xước) dù nó nằm ở bất kỳ vị trí nào trên vỏ lon.



Hình 1.2. *Minh họa phép tính tích chập trên ảnh.(Nguồn: Researchgate.net)*

Khi ảnh đầu vào được đưa qua lớp tích chập, các bộ lọc (filters) hoặc hạt nhân (kernels) sẽ quét qua toàn bộ ảnh với một bước nhảy (stride) nhất định. Mỗi bộ lọc thực hiện phép nhân giữa giá trị của các điểm ảnh trong một vùng nhỏ (gọi là Local Receptive Field - LRF) với các giá trị trọng số của bộ lọc, sau đó cộng lại để tạo ra một giá trị duy nhất trong bản đồ đặc trưng (feature map). Quá trình này được lặp đi lặp lại trên toàn bộ ảnh, với các bộ lọc khác nhau được sử dụng để học các đặc trưng khác nhau.

Một đặc điểm quan trọng của lớp tích chập là khả năng phát hiện các đặc trưng bất kể vị trí của chúng trong ảnh (invariance to spatial translation). Ví dụ, nếu một cạnh hoặc một góc xuất hiện ở bất kỳ vị trí nào trong ảnh, lớp tích chập vẫn có thể phát hiện ra nhờ các bộ lọc. Do đó, lớp tích chập được coi như một bộ trích xuất đặc trưng hiệu quả.

Ngoài ra, sau khi tích chập, kết quả thường được đưa qua một hàm kích hoạt phi tuyến như ReLU (Rectified Linear Unit), nhằm tăng khả năng biểu diễn của mạng bằng cách loại bỏ các giá trị âm và giữ nguyên các giá trị dương. Điều này giúp mạng xử lý tốt hơn các quan hệ phi tuyến trong dữ liệu.

b. Lớp gộp (Pooling Layer)

Lớp gộp [29] đóng vai trò giảm kích thước của bản đồ đặc trưng trong khi vẫn bảo toàn các đặc trưng quan trọng nhất. Điều này giúp giảm số lượng tham số trong mạng, giảm độ phức tạp tính toán, và tránh hiện tượng overfitting. Quá trình gộp thường được thực hiện trên các vùng cục bộ của bản đồ đặc trưng, tương tự như trong lớp tích chập.

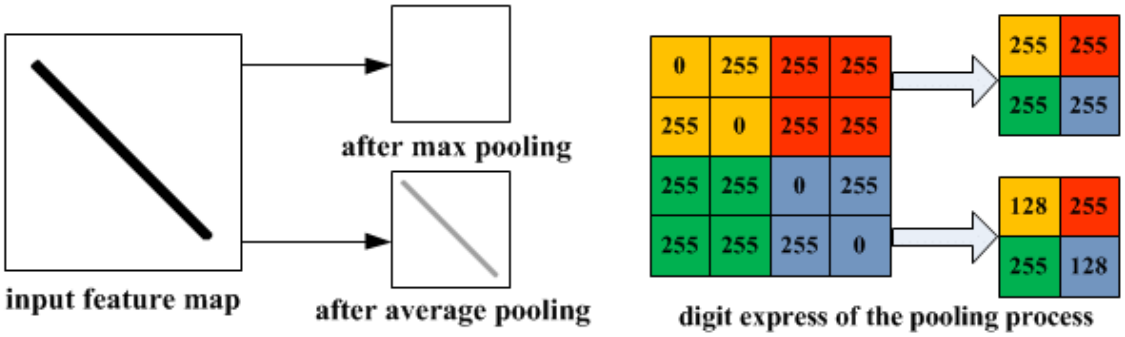
Có hai phương pháp gộp phổ biến:

MaxPooling: Lấy giá trị lớn nhất trong mỗi vùng cục bộ, giúp mạng giữ lại các đặc trưng nổi bật nhất.

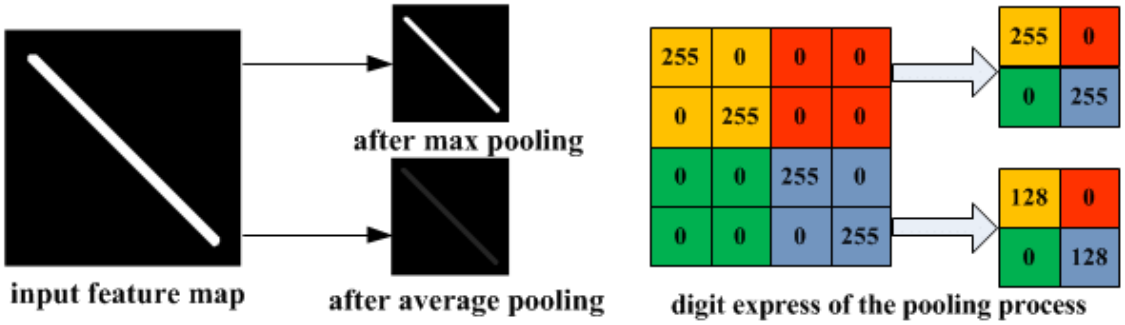
AveragePooling: Lấy giá trị trung bình của các điểm ảnh trong vùng cục bộ, mang lại thông tin tổng quát hơn.

Ví dụ, nếu bản đồ đặc trưng có kích thước lớn, việc áp dụng lớp gộp có thể giảm kích thước của nó xuống chỉ còn một phần nhỏ, trong khi vẫn giữ được các thông tin quan trọng cho quá trình học.

Lớp gộp không có tham số học mà hoạt động như một công cụ giảm độ phức tạp của dữ liệu. Ngoài ra, nó giúp tăng tính bền vững của mạng đối với các biến dạng nhỏ trong dữ liệu đầu vào, chẳng hạn như thay đổi kích thước hoặc xoay ảnh [30].



(a) Illustration of max pooling drawback

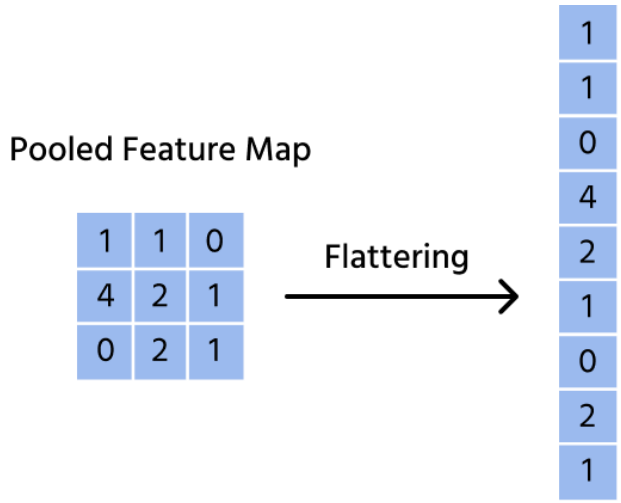


(b) Illustration of average pooling drawback

Hình 1.3. *Minh họa kết quả của một feature map sau khi qua MaxPooling và AveragePooling. (Nguồn: Hanli Wang)*

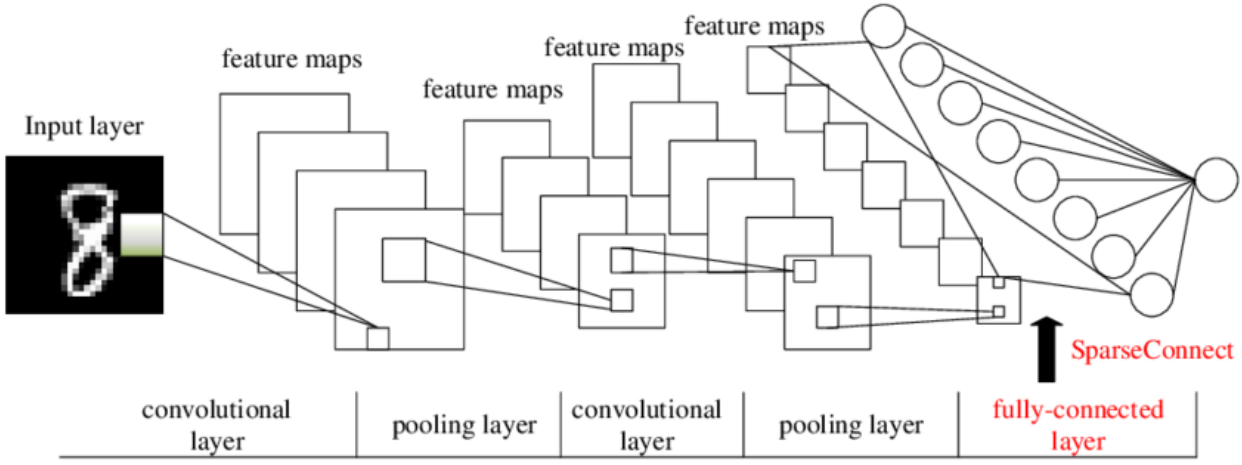
c. Lớp kết nối đầy đủ (Fully Connected Layer - FC)

Lớp kết nối đầy đủ (Fully Connected Layer - FC) là phần cuối cùng trong kiến trúc CNN, nơi tất cả các nơ-ron trong lớp hiện tại được kết nối đầy đủ với tất cả các nơ-ron trong lớp tiếp theo. Vai trò chính của lớp này là sử dụng các đặc trưng đã được trích xuất từ các lớp trước đó để thực hiện các nhiệm vụ cụ thể như phân loại hoặc dự đoán.



Hình 1.4. *Trải dài ma trận bằng lớp Flattening. (Nguồn: codefinity)*

Trong bài toán phân loại, đầu ra của lớp này thường là một vector biểu diễn xác suất các lớp. Chẳng hạn, trong bài toán phân loại nhị phân, lớp FC sẽ có một nơ-ron đầu ra duy nhất với hàm kích hoạt Sigmoid để biểu diễn xác suất thuộc về một lớp cụ thể. Trong bài toán phân loại đa lớp, số lượng nơ-ron đầu ra bằng số lớp cần phân loại, và hàm kích hoạt Softmax được sử dụng để chuẩn hóa các giá trị đầu ra thành xác suất.



Hình 1.5. *Minh họa lớp kết nối đầy đủ. (Nguồn: Qi Xu)*

Một điểm khác biệt quan trọng giữa lớp kết nối đầy đủ trong CNN và mạng nơ-ron truyền thống là kích thước của dữ liệu đầu vào. Dữ liệu đầu vào của lớp kết nối đầy đủ trong CNN đã được giảm kích thước đáng kể qua các lớp tích chập và gộp, giúp giảm

đáng kể số lượng tham số và độ phức tạp tính toán so với mạng truyền thống. Điều này làm cho CNN trở nên hiệu quả hơn trong việc xử lý dữ liệu lớn như hình ảnh.

1.3.2. Các hàm kích hoạt phi tuyến (Activation Functions)

Lớp kích hoạt trong một mạng nơ-ron đóng vai trò quan trọng trong việc đưa ra các quyết định phi tuyến tính, giúp mạng học các mối quan hệ phức tạp trong dữ liệu. Nếu không có lớp kích hoạt, các mạng nơ-ron chỉ có thể thực hiện các phép toán tuyến tính, điều này sẽ giới hạn khả năng mô hình hóa các chức năng phức tạp. Lớp kích hoạt thường được áp dụng sau mỗi phép toán ở các lớp trong mạng nơ-ron, giúp mạng có thể học các đặc trưng phi tuyến của dữ liệu.

Các hàm kích hoạt phổ biến:

- **Sigmoid:** Hàm sigmoid có dạng $f\ x\ = \frac{1}{1+e^{-x}}$, giúp chuyển đổi giá trị đầu vào thành một giá trị trong khoảng từ 0 đến 1. Hàm này thường được sử dụng trong các bài toán phân loại nhị phân, vì nó cho phép mô hình đưa ra xác suất cho từng lớp.
- **ReLU (Rectified Linear Unit):** Hàm ReLU có dạng $f(x) = \max(0, x)$, và thường được sử dụng trong các mạng nơ-ron hiện đại, đặc biệt là trong các mô hình học sâu. ReLU giúp giải quyết vấn đề vanishing gradient và tăng tốc độ hội tụ của mô hình trong quá trình huấn luyện.
- **Tanh (Hyperbolic Tangent):** Hàm tanh có dạng $f\ x\ = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$, chuyển đổi giá trị đầu vào thành một giá trị trong khoảng từ -1 đến 1. Tanh thường được sử dụng trong các ứng dụng yêu cầu dữ liệu đầu ra có giá trị đối xứng, nhưng đôi khi nó cũng gặp phải vấn đề vanishing gradient.
- **Leaky ReLU:** Để khắc phục vấn đề dying ReLU, Leaky ReLU đã được giới thiệu. Công thức của Leaky ReLU là: $L\text{eakyReLu}(x) = \max(ax, x)$. Với α là một hằng số nhỏ, giúp tránh các nơ-ron chết trong mạng.

- **Softmax:** Softmax là một hàm kích hoạt đặc biệt được sử dụng chủ yếu trong lớp đầu ra của các bài toán phân loại đa lớp. Nó chuyển đổi đầu ra của các nơ-ron trong lớp cuối thành xác suất cho mỗi lớp, với tổng xác suất bằng 1. Hàm Softmax có dạng: $p_i = \frac{e^{z_i}}{\sum_{j-1}^{n} e^{z_i}}$, với e^{z_i} là giá trị hàm mũ z_i. Tổng ở mẫu số là tổng của các giá trị hàm mũ của tất cả các phần tử trong vector đầu vào.

Hiện nay, hàm kích hoạt được sử dụng phổ biến nhất là hàm ReLU do tính hiệu quả cao của nó. Hàm này được áp dụng ngay sau bước tính toán tích chập, tạo ra một feature map mới có cùng kích thước với feature map đầu vào. Điểm khác biệt chính là các giá trị âm trong feature map đầu vào đã được loại bỏ, trong khi các giá trị khác vẫn được giữ nguyên.

1.3.3. Hàm mất mát (Loss Functions)

Hàm mất mát (*loss function*) đóng vai trò quan trọng trong việc huấn luyện mạng nơ-ron tích chập (CNN) cho các bài toán phân loại.

Binary Cross-Entropy (BCE) - còn gọi là Log Loss, là một hàm mất mát dùng trong phân loại nhị phân. Đo lường sự khác biệt giữa hai phân phối xác suất: phân phối thật $y \in \{0,1\}$ và phân phối dự đoán $\hat{y} \in [0,1]$. Giá trị BCE càng nhỏ thì mô hình càng dự đoán gần đúng với nhãn thực.

Công thức toán học:

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \quad (1.2)$$

Phân tích:

- Nếu nhãn thực $y = 1$ (OK): Mô hình bị phạt nặng nếu dự đoán $\hat{y}$ gần 0.
- Nếu nhãn thực $y = 0$ (NG): Mô hình bị phạt nặng nếu dự đoán $\hat{y}$ gần 1.

Ưu điểm: Phạt rất nặng các dự đoán sai với độ tin cậy cao, giúp mô hình hội tụ nhanh về trạng thái phân loại chính xác.

Mean Squared Error (MSE) hay còn gọi là Sai số bình phương trung bình

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2 \quad (1.3)$$

Trong đó: x_i là ảnh đầu vào gốc, $\hat{x}_i$ là ảnh tái tạo và N là tổng số pixel.

1.4. Kiến trúc Autoencoder (AE)

Autoencoder (AE) là một loại mạng nơ-ron truyền thẳng được thiết kế theo kiến trúc học không giám sát (Unsupervised Learning), với chức năng chính là học biểu diễn mã hóa hiệu quả của dữ liệu đầu vào [32], [33].

1.4.1. Kiến trúc cơ bản

Kiến trúc của AE bao gồm hai phần chính được kết nối và đối xứng với nhau:

- **Encoder (Bộ mã hóa):** Đây là phần chịu trách nhiệm nén dữ liệu đầu vào thành một biểu diễn có số chiều thấp hơn, gọi là không gian tiềm ẩn (Latent Space Representation). Trong các AE tích chập (CAE) được sử dụng cho ảnh, Encoder bao gồm các lớp Conv2D kết hợp với MaxPooling2D để trích xuất đặc trưng và giảm kích thước không gian.
- **Decoder (Bộ giải mã):** Decoder thực hiện nhiệm vụ tái tạo lại dữ liệu đầu vào từ không gian tiềm ẩn. Nó thường sử dụng các lớp Conv2D kết hợp với UpSampling2D (hoặc Conv2DTranspose) để tăng kích thước không gian trở về kích thước ban đầu.

1.4.2. Cơ chế phát hiện bất thường

Trong các bài toán công nghiệp, khi dữ liệu lỗi (NG) rất hiếm và khó thu thập đầy đủ các biến thể , AE được ứng dụng để phát hiện bất thường.

Nguyên lý: Mô hình được huấn luyện chỉ bằng các mẫu dữ liệu bình thường. Khi gặp một mẫu lỗi, mô hình không thể tái tạo chính xác các vết lỗi, dẫn đến sai số tái tạo (Reconstruction error) cao. Bất kỳ mẫu nào có sai số tái tạo vượt qua một ngưỡng xác định sẽ được coi là bất thường.

Hàm mất mát: Để tối ưu hóa quá trình tái tạo, hàm mất mát phổ biến là Sai số bình phương trung bình (Mean Squared Error - MSE):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2 \quad (1.4)$$

Trong đó: x_i là ảnh đầu vào gốc, $\hat{x}_i$ là ảnh tái tạo và N là tổng số pixel.

1.5. Mô hình EfficientNet

EfficientNet [34] là một kiến trúc mạng nơ-ron tích chập (CNN) hiện đại, được phát triển để đạt hiệu suất cao nhất trong các bài toán phân loại hình ảnh (Image Classification).

1.5.1. Nguyên tắc Compound Scaling

EfficientNet nổi bật nhờ nguyên tắc Compound Scaling, sử dụng một hệ số nhân tổng hợp để tăng đồng thời ba chiều của mạng một cách đồng bộ:

- Chiều rộng (Width): Số lượng kênh (filters) của lớp.
- Chiều sâu (Depth): Số lượng lớp (layers) của mạng.
- Độ phân giải (Resolution): Kích thước ảnh đầu vào.

Việc điều chỉnh đồng bộ này giúp EfficientNet duy trì sự cân bằng tối ưu giữa các tài nguyên của mô hình, từ đó đạt được độ chính xác vượt trội với chi phí tính toán thấp hơn so với các mô hình CNN lớn khác.

1.5.2. Ứng dụng học chuyển giao (Transfer Learning)

Trong thực nghiệm, **EfficientNetB0** được sử dụng theo phương pháp học chuyển giao (Transfer Learning):

- Mô hình được tải với trọng số đã huấn luyện trước trên tập dữ liệu ImageNet.
- Lớp phân loại mới: Lớp đầu phân loại gốc được loại bỏ include_top=False và thay thế bằng các lớp tùy chỉnh (như GlobalAveragePooling2D và Dense).
- Phân loại nhị phân: Lớp Dense cuối cùng sử dụng hàm kích hoạt Sigmoid để trả về xác suất thuộc lớp đạt tiêu chuẩn.
- Hàm mất mát: Sử dụng Binary Cross-Entropy (BCE) để tối ưu hóa quá trình phân loại nhị phân

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \quad (1.5)$$

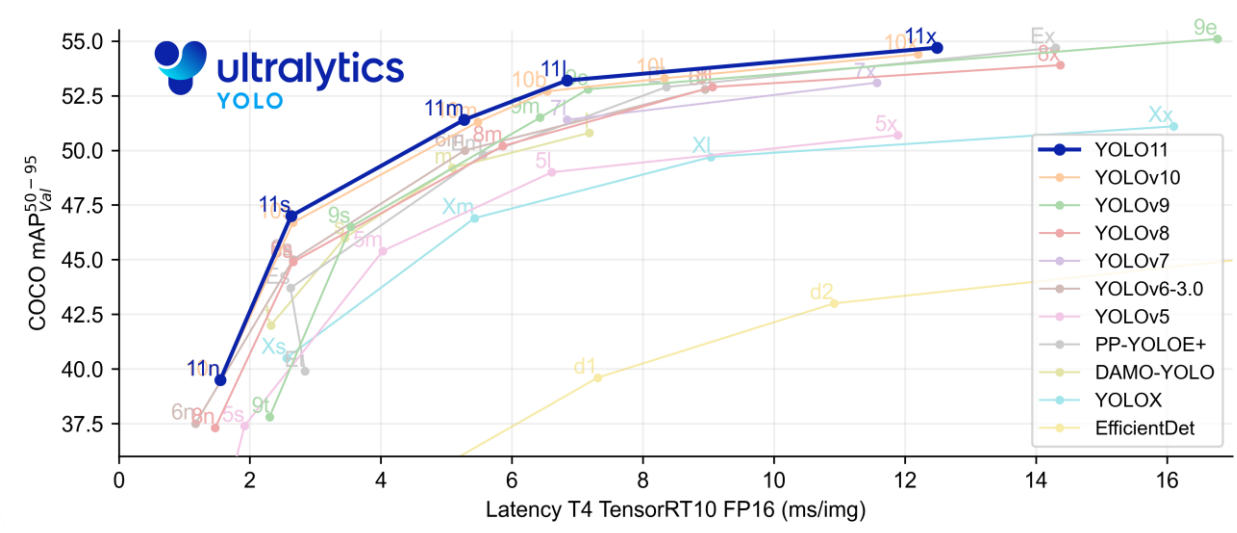
1.6. Mô hình YOLO (You Only Look Once)

YOLO [35], [36], viết tắt của "You Only Look Once", là một trong những mô hình phát hiện đối tượng và phân đoạn hình ảnh nổi bật trong lĩnh vực thị giác máy tính. Được giới thiệu lần đầu vào năm 2015 bởi Joseph Redmon và Ali Farhadi, YOLO nhanh chóng khẳng định vị thế của mình nhờ vào tốc độ xử lý nhanh và độ chính xác cao, điều này làm cho nó trở thành lựa chọn phổ biến cho nhiều ứng dụng thực tiễn.

Từ phiên bản đầu tiên, YOLO đã trải qua nhiều cải tiến đáng kể qua các phiên bản tiếp theo. YOLOv2, ra mắt vào năm 2016, đã nâng cao độ chính xác bằng cách áp dụng các kỹ thuật như chuẩn hóa theo lô và hộp neo. Phiên bản YOLOv3, phát hành năm 2018, sử dụng kiến trúc mạng xương sống hiệu quả hơn, cho phép phát hiện đối tượng ở nhiều kích thước khác nhau.

Năm 2020 đánh dấu sự ra mắt của YOLOv4, với những cải tiến về tăng cường dữ liệu và các thuật toán phát hiện mới. YOLOv5, cũng được phát hành cùng năm, tập trung vào việc đơn giản hóa quy trình sử dụng và cải thiện hiệu suất tổng thể. Các phiên bản sau này, như YOLOv6 và YOLOv7, không chỉ tối ưu hóa hiệu suất mà còn mở rộng khả năng ứng dụng trong các lĩnh vực như robot giao hàng và ước tính tư thế.

Mới đây, YOLOv8 và các phiên bản tiếp theo như YOLOv9, YOLOv10, và YOLOv11(mới nhất) đã tiếp tục cải tiến hiệu suất và tính linh hoạt, hỗ trợ đầy đủ cho nhiều tác vụ AI về thị giác, từ phát hiện đối tượng đến phân đoạn và theo dõi.



Hình 1.6. *COCO mAP của từng phiên bản YOLO (Nguồn: ultralytics)*

YOLO hoạt động dựa trên một mạng nơ-ron tích chập (CNN) được thiết kế đặc biệt, bao gồm ba phần chính: backbone, neck và head.

- Backbone: Đây là phần cốt lõi của mạng, có nhiệm vụ trích xuất các đặc trưng sâu từ hình ảnh đầu vào. Các đặc trưng này sẽ được sử dụng để xác định các đối tượng trong hình ảnh. Các kiến trúc backbone phổ biến trong YOLO bao gồm Darknet, EfficientNet,...

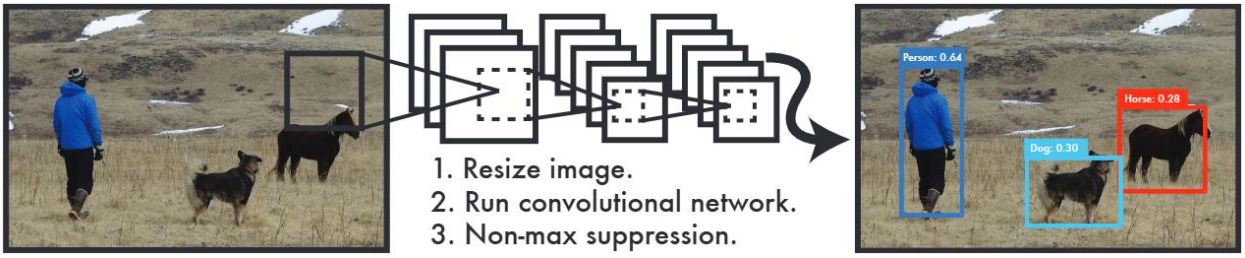
- Neck: Sau khi trích xuất đặc trưng, hình ảnh sẽ được truyền qua phần neck của mạng. Neck có nhiệm vụ kết hợp các đặc trưng từ các lớp khác nhau của backbone để tạo ra một biểu diễn đặc trưng phong phú hơn, giúp mạng có khả năng phát hiện các đối tượng ở nhiều kích thước và tỷ lệ khác nhau.

- Head: Đây là phần cuối cùng của mạng, chịu trách nhiệm dự đoán các thông tin về các đối tượng trong hình ảnh. Cụ thể, head sẽ dự đoán:

- Chia hình ảnh thành lưới (Grid): Hình ảnh đầu vào được chia thành một lưới S × S, trong đó mỗi ô lưới chịu trách nhiệm dự đoán đối tượng trong ô đó.
- Dự đoán bounding boxes và confidence scores: Mỗi ô lưới sẽ dự đoán một số lượng hộp chứa giới hạn kèm theo độ tin cậy confidence. Độ tin cậy được tính bằng:

$$\text{confidence} = \text{P(object)} \times \text{IOU} \quad (1.6)$$

- Lựa chọn và loại bỏ hộp dư thừa (Non-maximum Suppression – NMS): YOLO áp dụng thuật toán NMS để loại bỏ các hộp dự đoán dư thừa và chỉ giữ lại hộp có độ tin cậy cao nhất cho mỗi đối tượng.



Hình 1.7. *Minh hoạ hệ thống xử lý hình ảnh YOLO (Nguồn: arxiv)*

YOLO được ứng dụng rộng rãi trong nhiều lĩnh vực, bao gồm: giám sát an ninh, lái xe tự động, nhân diện đồ vật, y tế,.... YOLO là một phương pháp đột phá trong lĩnh vực nhận diện đối tượng, với sự kết hợp giữa tốc độ và độ chính xác cao. Qua nhiều phiên bản cải tiến, YOLO không chỉ khẳng định được tính hiệu quả mà còn đặt nền móng cho các nghiên cứu và ứng dụng sâu rộng trong các bài toán thị giác máy tính.

1.7. Công cụ sử dụng

1.7.1. Ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình bậc cao, đa năng, được Guido van Rossum tạo ra và ra mắt lần đầu vào năm 1991 [38]. Python nổi tiếng với cú pháp rõ ràng, dễ đọc, dễ học và dễ nhớ, sử dụng thụt đầu dòng để phân chia khối lệnh, cho phép người dùng viết mã với số lần gõ phím tối thiểu [39]. Python được ứng dụng rộng rãi trong nhiều lĩnh vực, bao gồm phát triển web (với các framework như Django [40], Flask [41]), khoa học dữ

liệu (với các thư viện như NumPy [42], Pandas [43]) và học máy (với các framework như scikit-learn [44], TensorFlow [45], PyTorch [46]).

Sự phát triển của Python đến nay có thể chia làm ba giai đoạn chính: Python 1, Python 2 và Python 3. Tuy nhiên, Python 3 không hoàn toàn tương thích ngược với Python 2, và Python 2 đã chính thức ngừng hỗ trợ từ năm 2020 [47]. Sau 30 năm lãnh đạo cộng đồng, Guido van Rossum đã từ chức Leader vào tháng 7 năm 2018. Cộng đồng Python đóng vai trò quan trọng trong việc phát triển ngôn ngữ, bao gồm việc đóng góp mã nguồn, viết tài liệu và hỗ trợ người dùng.

Kể từ khi ra mắt Python 3.0 vào năm 2008, ngôn ngữ này đã trải qua nhiều phiên bản cập nhật, mỗi phiên bản đều mang đến những cải tiến đáng kể. Python 3.5 giới thiệu async/await cho lập trình bất đồng bộ và type hints cho kiểm tra kiểu tĩnh. Python 3.6 cải thiện f-string và chú thích biến. Python 3.7 bổ sung dataclasses. Python 3.8 thêm toán tử walrus và cải thiện typing. Các phiên bản tiếp theo như 3.9, 3.10, 3.11 và 3.12 tiếp tục tập trung vào việc cải thiện hiệu năng, bổ sung tính năng mới và tăng cường bảo mật [48].

1.7.2. Roboflow

Roboflow là một nền tảng cho phép người sử dụng phát triển những mô hình thị giác máy tính riêng cho mình. Bằng cách cung cấp các công cụ cần thiết để triển khai từ ý tưởng đến xây dựng những mô hình thị giác máy tính mạnh mẽ [49].

Roboflow hỗ trợ quản lý dữ liệu và cung cấp các công cụ mạnh mẽ để gắn nhãn hình ảnh cho các tác vụ như phát hiện đối tượng, phân loại hình ảnh và phân đoạn ngữ nghĩa. Nền tảng này còn cho phép tăng cường dữ liệu huấn luyện bằng cách áp dụng các kỹ thuật như xoay, lật và thay đổi độ sáng, đồng thời chuyển đổi dữ liệu hình ảnh sang định dạng phù hợp với nhiều mô hình khác nhau.

1.7.3. Google Colab

Colab (hay còn gọi là “Colaboratory”) là một sản phẩm của Google Research, cho phép người dùng viết và thực thi Python trong trình duyệt và đặc biệt phù hợp với

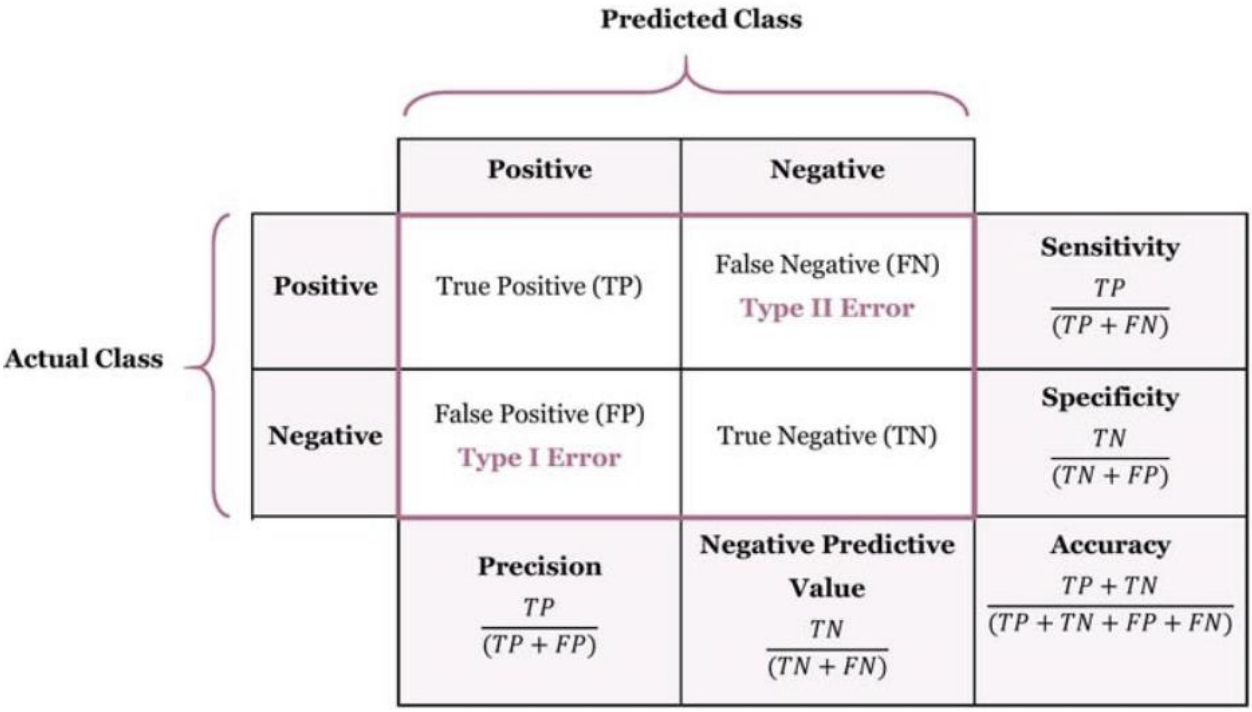
machine learning, phân tích dữ liệu và giáo dục với các lợi ích sau: không yêu cầu cấu hình, quyền truy cập miễn phí vào GPU, chia sẻ dễ dàng [50].

1.7.4. Ultralytics

Ultralytics [36] là một gói hỗ trợ được cung cấp bởi nhóm Ultralytics- nổi tiếng với việc phát triển YOLOv5, YOLOv8 và mới đây nhất là YOLO11. Ultralytics cung cấp giải pháp đơn giản hóa các quy trình: huấn luyện, kiểm thử và triển khai. Hỗ trợ thao tác với đa dạng các mô hình thị thị giác máy tính thuộc họ YOLO và nhiều mô hình nổi tiếng khác [51].

1.8. Các phương pháp đánh giá mô hình thị giác máy tính

1.8.1. Ma trận nhầm lẫn



Hình 1.8. *Ma trận nhầm lẫn (Nguồn: bigdatauni)*

Ma trận nhầm lẫn là một công hữu ích trong việc hỗ trợ và đánh giá các mô hình học máy, ma trận này chứa thông tin gồm các giá trị: Dương tính thật (TP), âm tính thật (TN), dương tính giả (FP), và âm tính giả (FN) [53].

1.8.2. Precision và Recall

Precision là một chỉ số đánh giá mức độ chính xác của mô hình trong việc dự đoán các mẫu thuộc lớp positive (lớp dương tính). Precision được xác định bằng tỷ lệ giữa số lượng true positives (dương tính thực) và tổng số mẫu được mô hình dự đoán là positive, bao gồm cả true positives (dương tính thực) và false positives (dương tính giả).

$$Precision = \frac{TP}{TP + FP}. \tag{1.7}$$

Recall, còn được gọi là sensitivity hoặc true positive rate, là thước đo khả năng của mô hình trong việc phát hiện tất cả các mẫu thuộc lớp positive (lớp dương tính). Recall được tính bằng cách lấy số lượng True Positives (dương tính thực) chia cho tổng số mẫu thực sự thuộc lớp positive, bao gồm cả True Positives (dương tính thực) và False Negatives (âm tính giả).

$$Recall = \frac{TP}{TP + FN} \tag{1.8}$$

Các tham số TP, FP, FN ở công thứ (1.7) và (1.8) được biểu diễn rõ ràng thông qua ma trận nhầm lẫn [54].

1.8.3. F1 Score và Accuracy

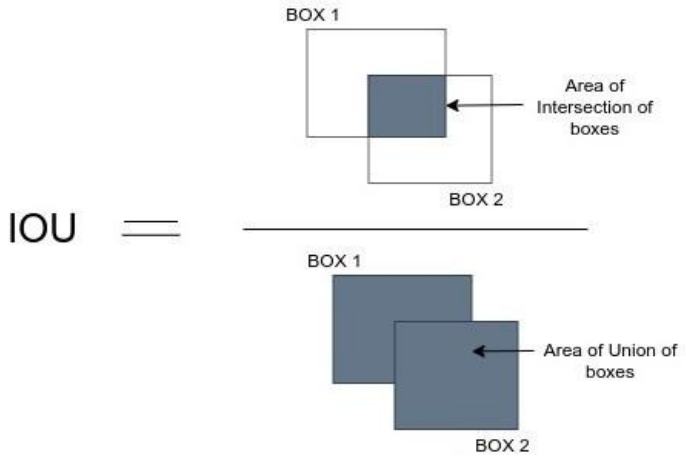
F1 Score – Chỉ số F1 là một thước đo hiệu suất dùng để đánh giá chất lượng của mô hình phân loại, đặc biệt hữu ích trong trường hợp dữ liệu bị mất cân bằng giữa các lớp (class imbalance). F1 Score là sự kết hợp hài hòa giữa hai chỉ số quan trọng là precision và recall, giúp cung cấp cái nhìn tổng quan về hiệu suất của mô hình. Giá trị F1 Score

càng cao thì đồng nghĩa với việc precision và recall đều cao, cho thấy mô hình hoạt động càng hiệu quả.

$$F1Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (1.9)$$

Accuracy – Độ chính xác là một trong những cách đơn giản nhất để đánh giá hiệu suất của mô hình phân loại. Chỉ số này được tính bằng tỷ lệ giữa số lượng mẫu được dự đoán đúng và tổng số mẫu trong tập dữ liệu kiểm thử.

1.8.4. IoU



Hình 1.9. *Minh họa các tham số cần để tính IoU. (Nguồn:viblo.asia)*

IoU là một thước đo để đánh giá mức độ trùng lặp giữa nhãn ô vuông dự đoán của mô hình so với nhãn thực tế của đối tượng. IoU = Diện tích phần giao nhau / Diện tích phần hợp.

1.8.5. AP và mAP

AP là diện tích dưới đường cong PR, thể hiện mối quan hệ giữa Precision và Recall khi thay đổi ngưỡng IoU [56], được tính toán như sau:

- Tính toán IoU: Đối với mỗi ô chữ nhật dự đoán, tính toán IoU với tất cả các ô chữ nhật là nhãn thực tế.

- Xác định TP, FP, và FN
- Tính toán Precision và Recall: Với mỗi lớp đối tượng, tính toán Precision và Recall cho các ngưỡng IoU khác nhau.
- Vẽ đường cong Precision-Recall: Vẽ đường cong PR cho mỗi lớp đối tượng.
- Tính toán AP: Tính diện tích dưới đường cong PR cho mỗi lớp.

Chỉ số mAP là trung bình cộng các giá trị AP của tất cả các lớp:

- mAP50: mAP được tính với ngưỡng IoU cố định là 0.5.
- mAP50-95: mAP được tính với các ngưỡng IoU từ 0.5 đến 0.95 với bước nhảy 0.05, sau đó tính trung bình cộng.

1.8.6. Đường cong AUC-ROC

AUC-ROC là một cặp số liệu đánh giá hiệu suất rất quan trọng được sử dụng để đo lường khả năng phân biệt của các mô hình phân loại nhị phân (binary classification models). Cụ thể: **ROC (Receiver Operating Characteristic)** Curve là một đường cong đồ thị minh họa sự đánh đổi giữa tỷ lệ dương tính đúng (True Positive Rate - TPR, còn gọi là độ nhạy/Sensitivity) và tỷ lệ dương tính giả (False Positive Rate - FPR) ở các ngưỡng phân loại khác nhau. **AUC (Area Under the Curve)** là diện tích nằm bên dưới đường cong ROC đó.

Ý nghĩa của AUC-ROC: AUC cung cấp một giá trị tóm tắt duy nhất cho hiệu suất của mô hình. Giá trị này luôn nằm trong khoảng từ 0 đến 1.

- AUC = 1: Mô hình có khả năng phân biệt hoàn hảo 100% giữa hai lớp (ví dụ: bệnh nhân bị bệnh và không bị bệnh).
- AUC = 0.5: Mô hình không có khả năng phân biệt; kết quả dự đoán của nó cũng ngẫu nhiên như việc tung đồng xu.

- AUC < 0.5: Mô hình hoạt động tệ hơn cả ngẫu nhiên (điều này cho thấy mô hình đang học sai quy luật và thường chỉ cần đảo ngược kết quả dự đoán là có thể cải thiện).

Nói đơn giản, giá trị AUC càng cao thì mô hình càng tốt trong việc phân loại các mẫu thuộc hai nhóm khác nhau. Nó thường được ưa dùng hơn các thước đo khác như độ chính xác (Accuracy) khi tập dữ liệu bị mất cân bằng (imbalanced dataset)

CHƯƠNG 2. XÂY DỰNG MÔ HÌNH THỰC NGHIỆM

2.1. Mô tả dữ liệu thực nghiệm

2.1.1. Bối cảnh và quy trình thu thập dữ liệu

Dữ liệu được thu thập trực tiếp tại nhà máy Carlsberg Việt Nam trong thời gian tác giả thực tập từ tháng 6 đến tháng 8. Hình ảnh được lấy từ hệ thống camera thuộc máy kiểm tra lon rỗng ECI (Empty Can Inspector) đặt ngay sau Depalletizer – đây là công đoạn đầu tiên khi lon rỗng được đưa vào dây chuyền.

Hệ thống camera này có nhiệm vụ chụp lại từng lon rỗng với góc nhìn top-view nhằm phát hiện các lỗi ngoại quan như méo, móp, deform viền lon, dị vật, bẩn hoặc phản xạ bất thường bên trong lon. Và cơ chế kỹ thuật lưu trữ dữ liệu bao gồm hai điểm quan trọng ảnh hưởng đến quá trình và thời gian thu thập dữ liệu:

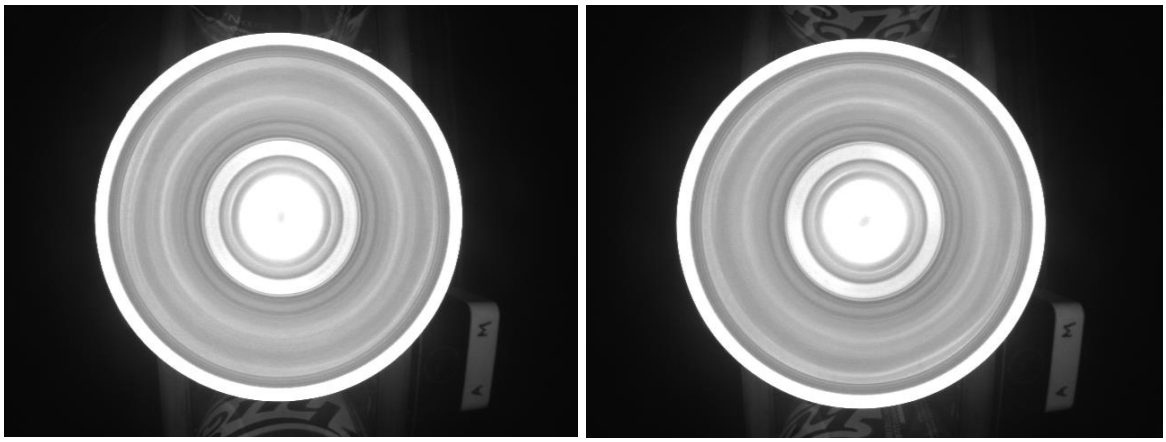
(1) Cơ chế lưu ảnh dạng buffer giới hạn: Máy chỉ giữ lại đúng 100 ảnh gần nhất cho mỗi loại (OK và NG). Khi ảnh mới được chụp, ảnh cũ sẽ bị ghi đè.→ Điều này ảnh hưởng đến thời gian thu thập dữ liệu vì không thể thu thập quá 200 ảnh trong một thời điểm được.

(2) Reset tên ảnh sau mỗi lần bắt đầu sản xuất: Mỗi khi dây chuyền khởi động, máy sẽ đặt lại số đếm trong tên file ảnh, giúp nhận ra thời điểm bắt đầu lô sản xuất và tránh lấy trùng dữ liệu, nhưng cũng đòi hỏi người thu thập phải ghi chú cẩn thận.

Ngoài ra, theo thực tế dây chuyền sản xuất bia, lon rỗng được nhà cung cấp kiểm định nghiêm ngặt trước khi giao cho công ty. Vì vậy, phần lớn lon đều đạt chuẩn (OK) và lon lỗi (NG) xuất hiện cực kỳ ít, chỉ xảy ra khi vận chuyển xảy ra va chạm hoặc lỗi vật liệu hiếm gặp. → Dữ liệu NG phải được thu thập trong thời gian kéo dài liên tục nhiều tuần mới đạt số lượng đủ phục vụ nghiên cứu.

Nhờ quá trình theo dõi liên tục tại dây chuyền, tổng cộng tác giả thu được: 3.000 ảnh lon đạt chuẩn OK; 1.616 ảnh lon không đạt chuẩn NG. Tổng số ảnh: 4.616. Toàn bộ dữ liệu đều ở định dạng BMP, kích thước ~302 KB và độ phân giải 640×480 px.

2.1.2. Đặc điểm kỹ thuật của hệ thống ghi nhận ảnh



Hình 2.10. *Minh hoạ ảnh lon rỗng thu được từ hệ thống*

Ảnh được chụp từ trên xuống (vuông góc với mặt phẳng miệng lon). Đặc điểm này mang lại nhiều lợi thế như: dễ quan sát độ tròn miệng lon (phát hiện méo, ovalization), có thể nhìn thấy phần bên trong lon hỗ trợ phát hiện dị vật, vết lõm hoặc bẩn và kiểm tra được các lỗi tại mép viền (rim) và thân lon gần miệng (Hình 2.10).

Chất lượng hình ảnh về độ sáng tương đối ổn định, đồng đều và không bị chói nặng. Có xuất hiện một số vùng phản xạ ánh sáng do bề mặt kim loại, nhưng mức độ không quá lớn. Độ tương phản giữa khu vực trong lon và viền lon tốt → thuận lợi cho segmentation.

2.1.3. Độ khó của dữ liệu và thách thức đối với mô hình học sâu

Trong quá trình áp dụng học sâu cho bài toán phân loại lon đạt và lon lỗi, dữ liệu đầu vào bộc lộ nhiều thách thức đặc thù. Trước hết, nhiều **lỗi có kích thước rất nhỏ** so với toàn bộ ảnh, khiến đặc trưng dễ bị hòa lẫn vào nền và khó được mô hình nhận diện. Điều này làm cho việc phân loại trở nên khó khăn, đồng thời các mô hình tái tạo như

autoencoder thường khôi phục lại hình ảnh mà không phân biệt được sự khác biệt giữa lon đạt và lon lỗi.

Một vấn đề khác là sự tương đồng giữa lon lỗi và lon đạt. Nhiều trường hợp, ngay cả con người khi quan sát nhanh cũng khó phát hiện, đặc biệt với các lỗi ở mép lon hoặc do phản xạ ánh sáng. Điều này đặt ra yêu cầu mô hình phải có khả năng học đặc trưng tinh vi và nhạy cảm với những biến đổi nhỏ.

Ngoài ra, tính không nhất quán của lỗi cũng gây khó khăn cho quá trình huấn luyện. Lỗi trên lon có thể khác nhau về hình dạng, kích thước, vị trí và mức độ rõ ràng, khiến mô hình segmentation khó tìm được một mẫu hình ổn định để học.

Cuối cùng, môi trường công nghiệp không cho phép kiểm soát tốt các điều kiện như ánh sáng, rung động hay góc chụp. Hệ quả là dữ liệu có thể sẽ bị nhòe nhẹ, lệch góc hoặc có vùng sáng tối không đồng nhất, làm tăng thêm độ khó cho việc huấn luyện và suy luận của mô hình học sâu.

2.2. Thực nghiệm với mô hình Autoencoder

Thực nghiệm này sử dụng mô hình Autoencoder để phát hiện các bất thường (lỗi) trên các sản phẩm lon. Autoencoder được huấn luyện chỉ bằng các ảnh sản phẩm đạt chuẩn (OK). Khi một ảnh lon lỗi (NG) được đưa vào, mô hình sẽ tái tạo kém, dẫn đến lỗi tái tạo - Reconstruction Error cao bất thường, cho phép phân loại ảnh đó là lỗi.

2.2.1. Chuẩn bị và xử lý dữ liệu

Bộ dữ liệu được nạp vào từ file nén DATA_FULL.zip trên Google Drive và giải nén vào thư mục /content/dataset. Tập dữ liệu huấn luyện: bao gồm 2732 ảnh lon đạt chuẩn OK từ tập dữ liệu thu thập được. Mục đích để huấn luyện mô hình học cách tái tạo chính xác các đặc trưng của một sản phẩm đạt chuẩn, từ đó thiết lập một chuẩn mực "bình thường" để có thể tìm ra điểm khác biệt của các sản phẩm lỗi.

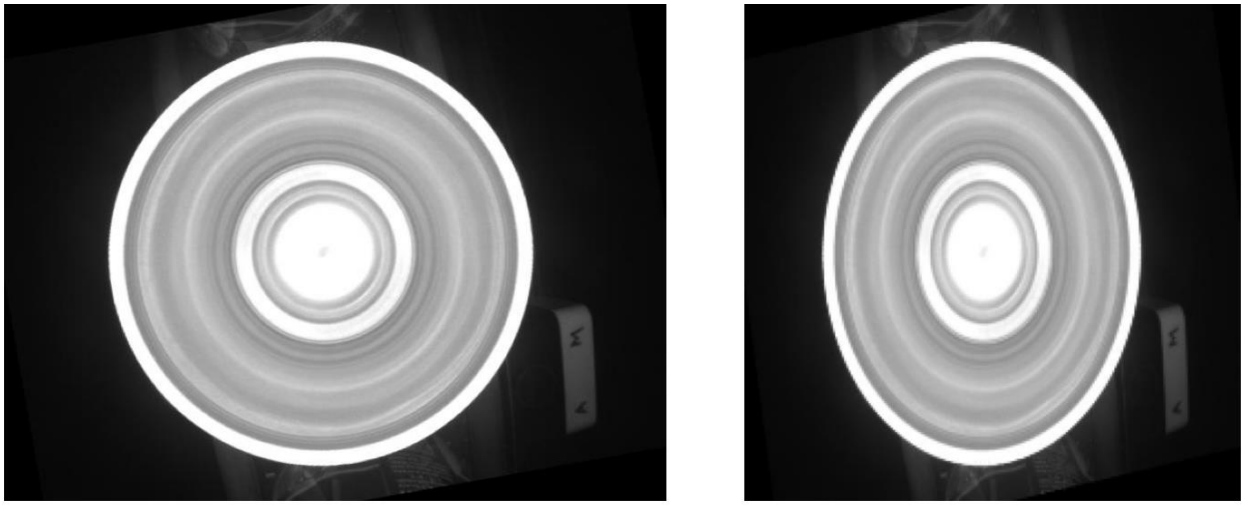
Quy trình tiền xử lý dữ liệu được áp dụng đồng nhất cho cả tập huấn luyện và tập kiểm tra, đảm bảo dữ liệu đầu vào phù hợp với mô hình học sâu bao gồm các bước:

- Đọc ảnh (Grayscale): Tất cả ảnh được đọc bằng thư viện OpenCV (cv2.imread) dưới định dạng ảnh xám (cv2.IMREAD_GRAYSCALE). Việc này giúp giảm kích thước dữ liệu (chỉ còn 1 kênh màu) và tăng tốc độ tính toán.

- Thay đổi kích thước: Kích thước ảnh đầu vào được đồng nhất về 224x224 pixels (target_size = (224, 224)) bằng hàm cv2.resize.

- Chuẩn hóa (Normalization): Giá trị pixel (có dải từ [0, 255]) được chuyển đổi sang kiểu dữ liệu float32 và chia cho 255.0. Thao tác này chuẩn hóa dải giá trị về khoảng [0.0, 1.0], một yêu cầu tiêu chuẩn cho hầu hết các mô hình mạng nơ-ron.

- Định hình lại (Reshape): Dữ liệu cuối cùng được chuyển thành mảng NumPy 4 chiều có dạng (Số lượng ảnh, Chiều cao, Chiều rộng, Kênh).



Hình 2.11. *Ảnh đầu vào và sau khi được xử lý để đưa vào mô hình Autoencoder (Bên trái: Ảnh đầu vào, bên phải: Ảnh sau khi được xử lý)*

2.2.2. Huấn luyện mô hình

Mô hình Autoencoder được xây dựng bằng API Sequential của TensorFlow/Keras, bao gồm hai phần chính: **Encoder** (Bộ mã hóa) và **Decoder** (Bộ giải mã). Với trình tối ưu

hoá (Optimizer): **adam**, hàm mất mát **mean_squared_error** (MSE). Mục tiêu là giảm thiểu MSE giữa ảnh gốc và ảnh tái tạo.

Các tham số huấn luyện:

- Số epoch: 50
- Batch size: 32
- Validation split: 0.30 (tức 30% của dataset_ok dùng cho validation trong quá trình huấn luyện)
- Shuffle=True

2.2.2.1. Encoder (Bộ mã hóa)

Encoder chịu trách nhiệm nén ảnh đầu vào có kích thước 224×224×1 thành một biểu diễn nén (Latent Space Representation). Chuỗi các lớp Conv2D với số filter tăng dần (32→64→128→256), mỗi block theo sau bởi một lớp MaxPooling2D (padding='same') để giảm kích thước không gian. Cuối encoder là không gian tiềm ẩn (latent) kích thước 14×14×256 (theo thiết kế pooling 3 lần từ input 128×128).

Layer (type)	Output Shape	Param #
input_layer_3 (InputLayer)	(None, 224, 224, 1)	0
conv2d_25 (Conv2D)	(None, 224, 224, 32)	320
max_pooling2d_11 (MaxPooling2D)	(None, 112, 112, 32)	0
conv2d_26 (Conv2D)	(None, 112, 112, 64)	18,496
max_pooling2d_12 (MaxPooling2D)	(None, 56, 56, 64)	0
conv2d_27 (Conv2D)	(None, 56, 56, 128)	73,856
max_pooling2d_13 (MaxPooling2D)	(None, 28, 28, 128)	0
conv2d_28 (Conv2D)	(None, 28, 28, 256)	295,168
max_pooling2d_14 (MaxPooling2D)	(None, 14, 14, 256)	0

Hình 2.12. *Kiến trúc của bộ mã hoá Encoder*

2.2.2.2. Decoder (Bộ giải mã)

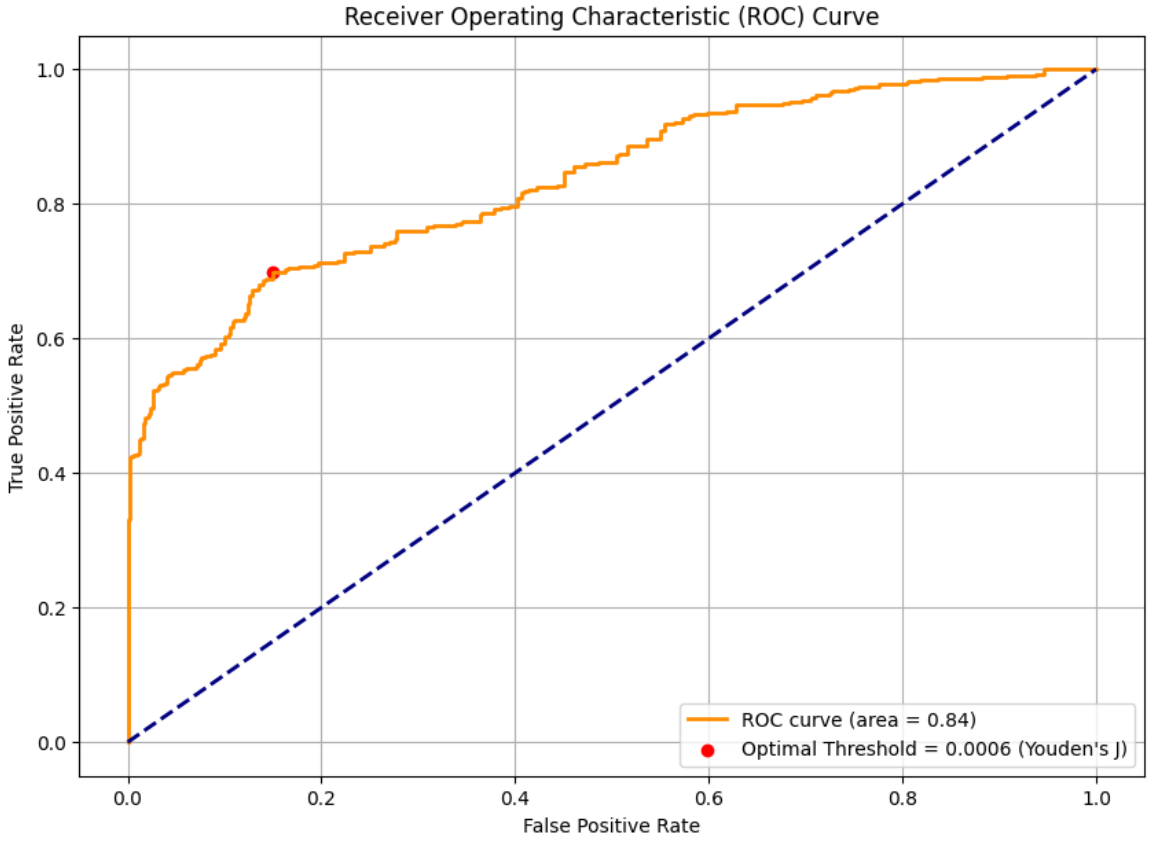
Decoder khôi phục lại ảnh từ biểu diễn nén, cố gắng tái tạo ảnh gốc. đối xứng với encoder: Conv2D + UpSampling2D lặp lại để đưa kích thước không gian trở về 128×128; lớp đầu ra là Conv2D(1, (3,3), activation='sigmoid') để đảm bảo output có giá trị trong [0,1]. Hàm kích hoạt sigmoid ở lớp cuối cùng đảm bảo giá trị pixel đầu ra nằm trong khoảng [0,1], khớp với ảnh đầu vào đã được chuẩn hóa.

conv2d_29 (Conv2D)	(None, 14, 14, 256)	590,080
up_sampling2d_11 (UpSampling2D)	(None, 28, 28, 256)	0
conv2d_30 (Conv2D)	(None, 28, 28, 128)	295,040
up_sampling2d_12 (UpSampling2D)	(None, 56, 56, 128)	0
conv2d_31 (Conv2D)	(None, 56, 56, 64)	73,792
up_sampling2d_13 (UpSampling2D)	(None, 112, 112, 64)	0
conv2d_32 (Conv2D)	(None, 112, 112, 32)	18,464
up_sampling2d_14 (UpSampling2D)	(None, 224, 224, 32)	0
conv2d_33 (Conv2D)	(None, 224, 224, 1)	289

Hình 2.13. *Kiến trúc của bộ giải mã Decoder*

2.2.2.3. Tối ưu ngưỡng bằng ROC và Youden's J

Để cải thiện hiệu suất của lớp "can_ng", thực hiện tối ưu hóa ngưỡng bằng cách xây dựng Đường cong ROC - Receiver Operating Characteristic để tính toán diện tích dưới đường cong AUC và chọn ngưỡng tối ưu bằng Youden's J (J = TPR - FPR). Kết quả ngưỡng tối ưu là 0.0006 với AUC = 0.84 cho thấy khả năng phân biệt hợp lý.



Hình 2.14. *Đồ thị ROC với điểm ngưỡng tối ưu được đánh dấu*

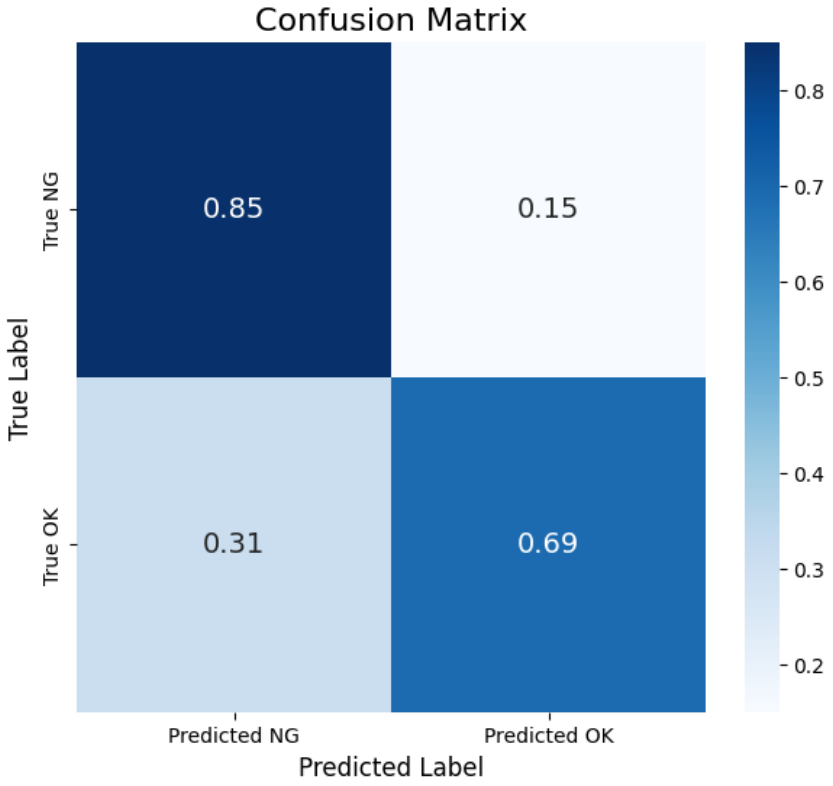
2.2.3. Kết quả

Bảng 2.1. *Kết quả mô hình Autoencoder*

Chỉ số	**Lon đạt chuẩn**	**Lon không đạt chuẩn**
Precision	0.73	0.82
Recall	0.85	0.69
F1-Score	0.79	0.75
Accuracy	68.80%	85.00%

Kết quả đánh giá mô hình Autoencoder được thực hiện trên tập kiểm thử, trong đó mô hình phân loại dựa trên **ngưỡng sai số tối ưu** (optimal threshold) là 0.0006 được xác định từ phân bố lỗi tái tạo (reconstruction error) trên tập validation đạt độ chính xác trung

bình 76.9%. Bảng kết quả các chỉ số đánh giá cho từng nhãn được biểu diễn ở Bảng 2.15 trên và ma trận hỗn hợp confusion matrix thu được thể hiện ở Hình 2.15.



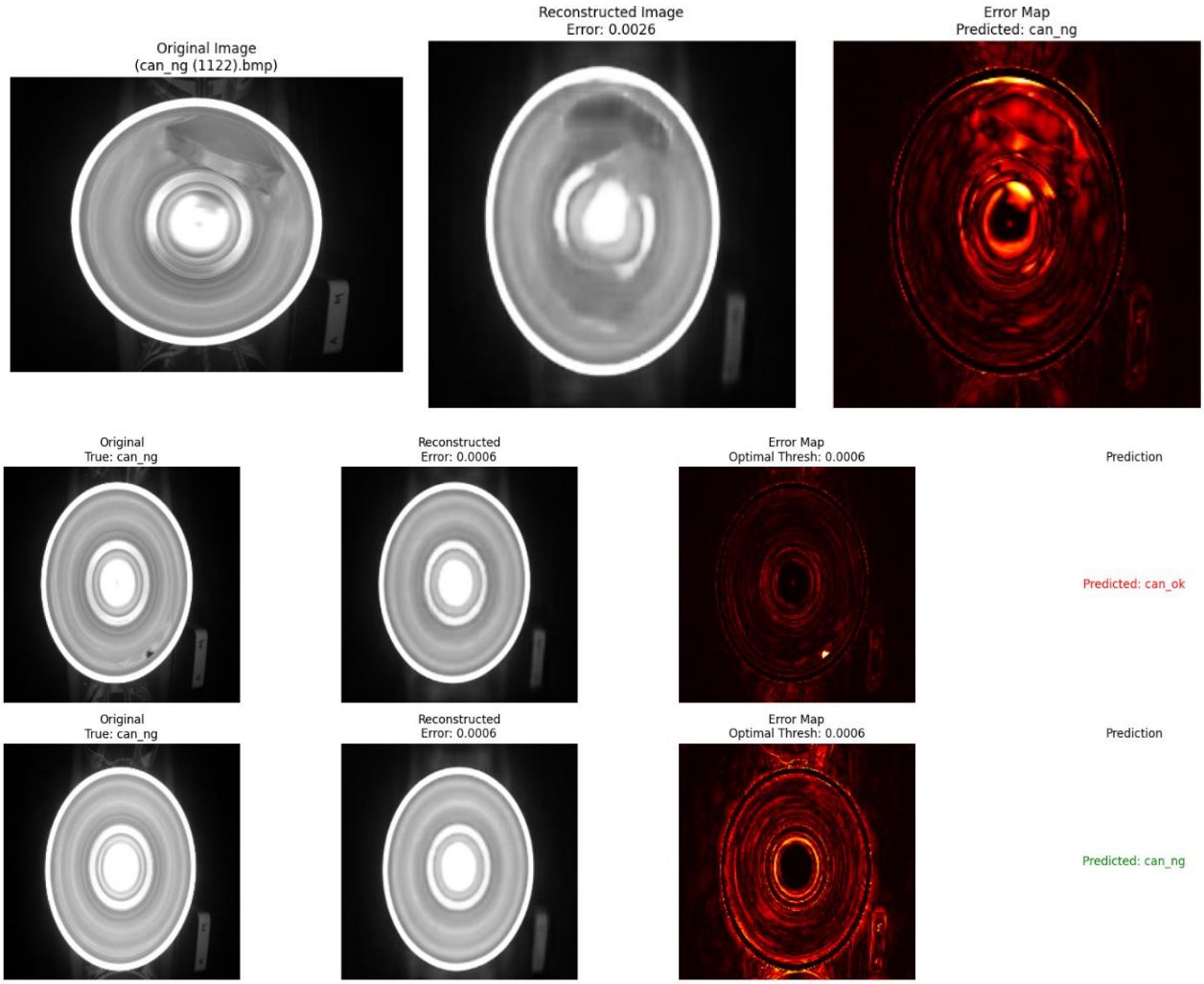
Hình 2.15. *Ma trận hỗn hợp của mô hình Autoencoder*

Dựa vào các kết quả đánh giá trên cho thấy Autoencoder hoạt động ở mức trung bình, phản ánh đúng đặc tính của bài toán anomaly detection với lỗi nhỏ và khó nhận biết. Với ngưỡng tối ưu được chọn, mô hình đạt Overall Accuracy = 76.90%, trong đó:

- Đối với lon đạt chuẩn (OK): mô hình nhận diện tốt với Accuracy = 85%, thể hiện khả năng tái tạo hiệu quả các mẫu bình thường.
- Đối với lon không đạt chuẩn (NG): mô hình gặp khó hơn, Accuracy chỉ đạt 68.8%, cho thấy nhiều lỗi nhỏ không tạo ra sai số tái tạo đủ lớn để vượt ngưỡng.

Chỉ số F1-score phản ánh tổng hợp cả Precision và Recall, với giá trị 0.79 cho lớp OK và 0.75 cho lớp NG. Điều này cho thấy mô hình hoạt động ổn định ở mức khá nhưng chưa đủ mạnh để áp dụng triển khai thực tế trong bối cảnh lỗi hiếm và khó nhận biết.

Kết quả dự đoán của mô hình Autoencoder được trực quan trong Hình 2.16, thể hiện quá trình xử lý một số mẫu lon qua các bước liên tiếp. Từ trái sang phải: đầu tiên là ảnh gốc đầu vào kèm nhãn thực tế; tiếp theo là ảnh tái tạo do mô hình sinh ra cùng giá trị sai số; sau đó là bản đồ sai số cho thấy mức độ khác biệt giữa ảnh gốc và ảnh tái tạo, trong đó các vùng màu sáng/đỏ biểu diễn khu vực sai lệch rõ rệt; từ đó cho ra kết quả phân loại mà mô hình đưa ra dựa trên ngưỡng sai số tối ưu. Hình ảnh này giúp người đọc quan sát trực tiếp cả trường hợp mô hình dự đoán đúng và trường hợp dự đoán sai, qua đó minh họa cách Autoencoder vận hành trong bài toán phát hiện bất thường.



Hình 2.16. *Minh hoạ kết quả dự đoán mô hình Autoencoder*

2.3. Thực nghiệm với mô hình EfficientNet

Thực nghiệm này sử dụng mô hình EfficientNetB0 đã được huấn luyện trước (Transfer Learning) để thực hiện bài toán Phân loại Nhị phân (Binary Classification) trực tiếp, nhằm phân loại ảnh đầu vào là Lon đạt chuẩn ('OK') hoặc Lon không đạt chuẩn ('NG'). Phương pháp này khác với Autoencoder ở chỗ nó học các đặc trưng phân biệt giữa hai lớp lỗi và không lỗi, thay vì chỉ học các đặc trưng của lớp bình thường.

2.3.1. Chuẩn bị và xử lý dữ liệu

Quá trình chuẩn bị dữ liệu đóng vai trò quan trọng trước khi huấn luyện mạng EfficientNet. Trong phần thực nghiệm, triển khai bốn bước chính: tải dữ liệu – tách bộ dữ liệu – tăng cường dữ liệu – chuẩn hóa và tối ưu hoá pipeline, dựa hoàn toàn trên đoạn mã của hệ thống.

Bộ dữ liệu được nén dưới dạng ZIP và được gắn vào Google Drive. Sau đó chương trình tự động giải nén vào thư mục /content/dataset gồm hai lớp: lon đạt chuẩn (OK) với 1366 ảnh và 1366 ảnh lon không đạt chuẩn (NG).

Dữ liệu được nạp bằng hàm tf.keras.utils.image_dataset_from_directory, tự động gán nhãn dựa theo tên thư mục con và tạo cấu trúc dataset dạng batch, thuận tiện cho huấn luyện mô hình EfficientNet.

Phân chia tập dữ liệu: 70% cho tập dữ liệu huấn luyện, 15% cho tập kiểm tra và 15% cho tập dự đoán.

Tăng cường Dữ liệu (Data Augmentation): Áp dụng các phép biến đổi ngẫu nhiên cho tập huấn luyện để tăng cường khả năng tổng quát hóa của mô hình. Các kỹ thuật được sử dụng bao gồm: RandomFlip('horizontal') - lật ngang ngẫu nhiên, RandomRotation(0.2) - oay ngẫu nhiên ±20% góc và RandomZoom(0.2) - phóng to/thu nhỏ ngẫu nhiên 20%. Các phép biến đổi này được áp dụng thông qua hàm augment() nhằm giảm overfitting và tăng khả năng tổng quát hóa mô hình.

Thay đổi kích thước và chuẩn hóa: Kích thước ảnh đầu vào được đồng nhất về 224×224 pixels. Sau đó, ảnh được chuẩn hóa giá trị pixel về dải [0.0, 1.0] bằng lớp tf.keras.layers.Rescaling(1./255.0)

Tối ưu pipeline dữ liệu (Caching & Prefetching): Để tăng tốc độ huấn luyện, tất cả các dataset đều áp dụng .cache() giúp lưu batch dữ liệu đầu tiên vào RAM, giảm thời gian tải lại và .prefetch(buffer_size=tf.data.AUTOTUNE) tải batch tiếp theo song song trong khi GPU xử lý batch hiện tại tối ưu hóa luồng dữ liệu, đảm bảo GPU không bị chờ I/O trong suốt quá trình đào tạo mô hình.

2.3.2. Huấn luyện mô hình

Mô hình EfficientNetB0 được sử dụng làm xương sống (backbone) cho bài toán phân loại:

Mô hình cơ sở (Base model): Tải mô hình EfficientNetB0 đã được huấn luyện trước trên tập ImageNet (weights='imagenet') mà không bao gồm lớp đầu phân loại (include_top=False). Đầu vào được thiết lập là ảnh RGB có kích thước (224,224,3).

Phần đầu phân loại gốc của ImageNet được loại bỏ (include_top=False) nhằm cho phép xây dựng tầng phân loại mới phù hợp với bài toán hiện tại. Các lớp phân loại mới vào sau backbone EfficientNet, bao gồm:

- GlobalAveragePooling2D: Giảm chiều không gian của tensor đầu ra từ EfficientNet bằng cách lấy trung bình toàn bộ bản đồ đặc trưng ở mỗi kênh.

- Dense(1, activation='sigmoid'): Tầng đầu ra với 1 nơ-ron và hàm kích hoạt Sigmoid, trả về xác suất thuộc lớp OK. Đây là cấu hình tiêu chuẩn cho bài toán phân loại nhị phân.

Quá trình huấn luyện được chia thành hai giai đoạn để tối ưu hiệu suất và giảm nguy cơ overfitting:

(1) Huấn luyện lớp đầu (đóng băng backbone): Lớp cơ sở base_model được đóng băng **base_model.trainable = False**. Chỉ các lớp tùy chỉnh mới thêm vào được huấn luyện. Mục đích: Điều chỉnh các lớp phân loại mới để chúng phù hợp với các đặc trưng đã học của EfficientNetB0 trên tập ImageNet. Sử dụng trọng số **Adam** với tốc độ học **learning_rate=0.001**, **loss=BinaryCrossentropy**. Kết quả thu được hiệu suất rất kém accuracy ≈50%, chỉ ra rằng các đặc trưng ImageNet cần được điều chỉnh thêm cho dữ liệu lon cụ thể.

(2) Tinh chỉnh toàn bộ mô hình (mở khóa backbone): Lớp cơ sở EfficientNetB0 được giải phóng **base_model.trainable = True**. Biên dịch lại với tốc độ học tập rất nhỏ **learning_rate=0.00001** mục tiêu là tinh chỉnh nhẹ nhàng các trọng số pre-trained, tránh phá vỡ những đặc trưng hữu ích đã được học từ ImageNet.

Thực hiện huấn luyện với 50 epoch, gọi **ModelCheckpoint** để lưu trữ trọng số tốt nhất **best_efficientnet_model.weights.h5** dựa trên val_loss. Sau 50 epoch tinh chỉnh, mô hình được đánh giá lại bằng cách tải trọng số tốt nhất đã lưu trong quá trình huấn luyện. Mô hình cuối cùng được đánh giá là mô hình sau giai đoạn tinh chỉnh với trọng số tốt nhất được tải lại.

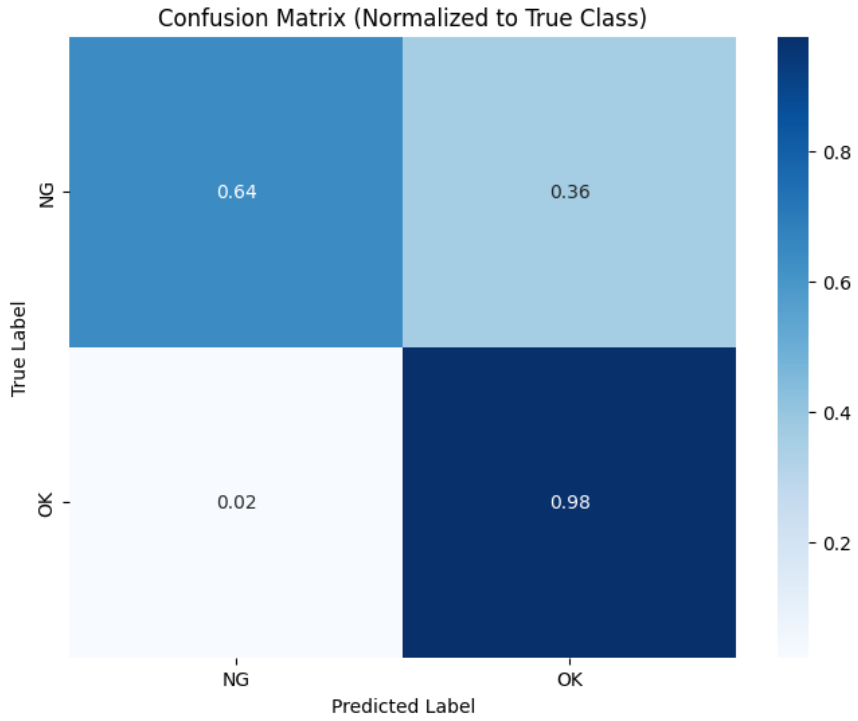
2.3.3. Kết quả

Mô hình Efficientnet sau khi huấn luyện, thu được kết quả các chỉ số đánh giá chi tiết tại Bảng 2.2 dưới đây.

Chỉ số	Giá trị
Precision	0.7267
Recall	0.9766
F1-Score	0.8333
Accuracy	80.62%

Bảng 2.2. *Kết quả huấn luyện của mô hình Efficientnet*

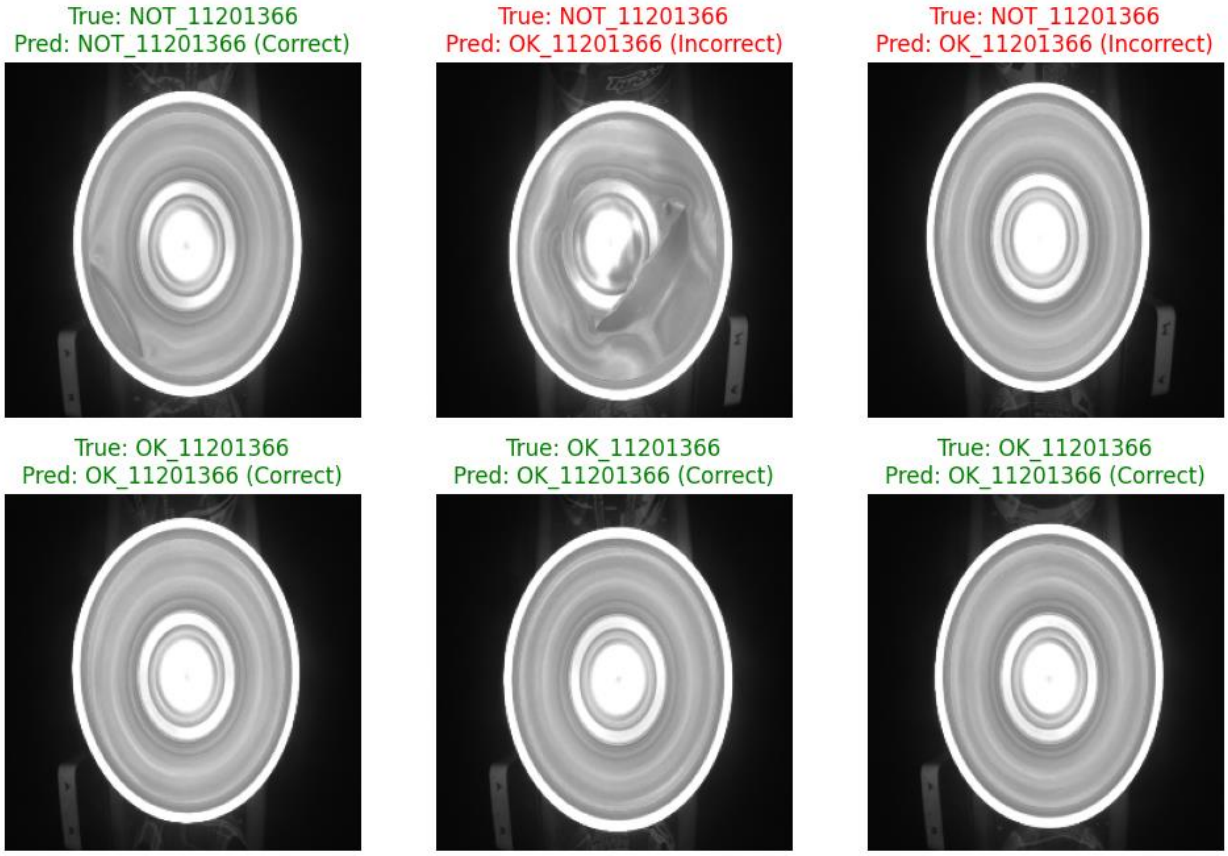
Nhìn vào bảng kết quả huấn luyện mô hình Efficientnet (Bảng 2.2) có thể thấy mô hình đạt hiệu suất cao với Recall 97.66% và F1-score 83.33%, cho thấy khả năng phát hiện đúng rất tốt. Tuy nhiên, Precision chỉ đạt 72.67%, nghĩa là vẫn còn tỷ lệ dự đoán sai NG tương đối cao.



Hình 2.17. *Kết quả ma trận nhầm lẫn của mô hình Efficientnet*

Ma trận nhầm lẫn (Hình 2.17) cho thấy mô hình phân loại rất tốt lớp OK (98% đúng), nhưng vẫn nhầm 36% mẫu NG thành OK. Điều này cho thấy mô hình thiên về độ nhạy, dễ bỏ sót lỗi NG nhẹ có thể gây ra rủi ro nghiêm trọng trong kiểm soát chất lượng.

Kết quả dự đoán của mô hình EfficientNet được trực quan hóa trong Hình 2.18, thể hiện khả năng phân loại của mô hình trên một số mẫu lon cụ thể. Mỗi ảnh đều đi kèm nhãn thực tế và kết quả dự đoán, qua đó giúp đánh giá trực tiếp độ chính xác của mô hình trong việc nhận diện đúng và sai giữa hai lớp OK và NG.



Hình 2.18. *Hình minh hoạ kết quả dự đoán mô hình EfficientNet*

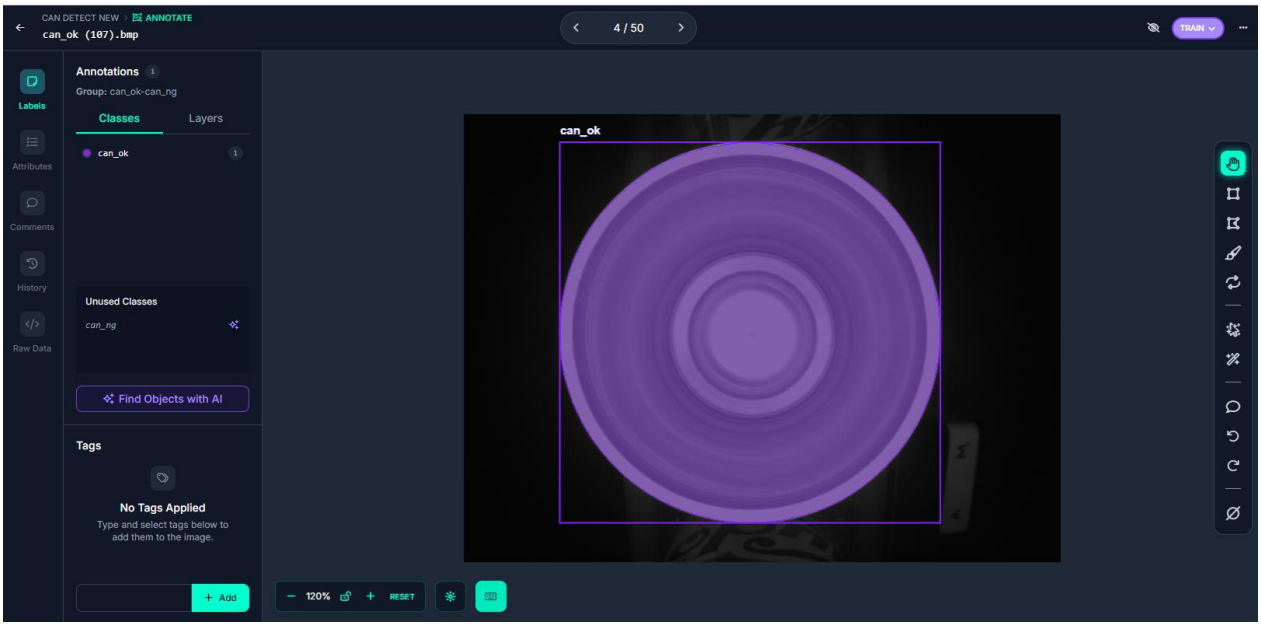
2.4. Thực nghiệm với mô hình YOLO

Thực nghiệm này sử dụng kiến trúc You Only Look Once (YOLO) để thực hiện tác vụ Phát hiện đối tượng (Object Detection) và Phân đoạn đối tượng (Instance Segmentation). Phương pháp này không chỉ phân loại ảnh ('OK'/'NG') mà còn xác định vị trí và hình dạng chính xác của đối tượng, cung cấp thông tin chi tiết cho kiểm soát chất lượng.

2.4.1. Chuẩn bị và xử lý dữ liệu

Sau khi hoàn tất việc thu thập dữ liệu, bắt đầu thực hiện quá trình gán nhãn dữ liệu, một bước quan trọng trong việc chuẩn bị tập dữ liệu để huấn luyện mô hình. Đầu tiên, các hình ảnh được tải lên Roboflow, một công cụ hỗ trợ gán nhãn hình ảnh mạnh mẽ và trực quan. Tiến hành gán nhãn thủ công bằng cách sử dụng các hộp giới hạn (bounding box)

và và mặt nạ phân đoạn (segmentation mask) để xác định vị trí và phạm vi của từng đối tượng trong ảnh. Trong quá trình gán nhãn, chú ý kiểm tra và loại bỏ những hình ảnh không đạt tiêu chuẩn, chẳng hạn như hình ảnh bị mờ, có độ nhiễu cao, hoặc khó xác định rõ ràng đối tượng. Việc này giúp giảm thiểu rủi ro overfitting và đảm bảo rằng chỉ những dữ liệu chất lượng cao mới được sử dụng trong quá trình huấn luyện mô hình.



Hình 2.19. *Minh hoạ quá trình gán nhãn bằng Roboflow*

Sau khi hoàn tất gán nhãn, tập dữ liệu thu được chứa tổng cộng 2732 nhãn đối tượng, với sự cân bằng hoàn hảo giữa hai lớp: 1366 nhãn lon đạt chuẩn can_ok và 1366 nhãn lon không đạt chuẩn can_ng. Tạo file data.yaml để cấu hình đường dẫn và xác định 2 lớp ['can_ng', 'can_ok'].

Phân chia tập dữ liệu: 70% cho tập dữ liệu huấn luyện được sử dụng để dạy mô hình nhận diện các đối tượng, 15% cho tập kiểm định (validation) nhằm đánh giá hiệu suất của mô hình trong suốt quá trình huấn luyện và điều chỉnh các tham số và 15% cho tập kiểm tra (test) dùng để kiểm tra độ chính xác và khả năng tổng quát hóa của mô hình sau khi hoàn thành huấn luyện.

Ngoài ra, khi gán nhãn trên Roboflow toàn bộ hình ảnh đều được chuẩn hóa về kích thước cố định 640x640 pixel để đảm bảo sự đồng nhất. Kích thước này không chỉ phù hợp với yêu cầu của YOLO mà còn tối ưu hóa hiệu suất trong quá trình huấn luyện. Nhờ vào các bước chuẩn bị kỹ lưỡng này, bộ dữ liệu tạo ra đảm bảo chất lượng cao, góp phần nâng cao độ chính xác và hiệu quả trong quá trình phát hiện và nhận diện đối tượng.

2.4.2. Huấn luyện mô hình

Sau khi hoàn tất quá trình xử lý, tiến hành tải tập dữ liệu đã được gán nhãn trên Roboflow để thực hiện huấn luyện, so sánh và chọn ra mô hình tối ưu cho bài toán phân loại lon. Tác giả sử dụng 2 mô hình là YOLOv11-seg (sử dụng trọng số yolo11n-seg.pt) và mô hình YOLOv8-seg (sử dụng trọng số yolo8n-seg.pt). Cả hai mô hình đều được huấn luyện với các tham số chung: epoch = 50, learning rate = 0.005, batch size = 16.

Quá trình huấn luyện sử dụng Adam optimizer kết hợp với Cross-Entropy Loss và IoU-based Loss để tối ưu hóa mô hình. Đây là các thuật toán cơ bản được duy trì trong cả ba phiên bản YOLO, giúp tối ưu việc dự đoán bounding box và phân loại chính xác các đối tượng. Ngoài ra, sử dụng Learning Rate Warm-up giúp điều chỉnh tốc độ học trong suốt quá trình huấn luyện để đạt hiệu quả tốt nhất. Mô hình được kiểm tra trên tập validation sau mỗi epoch để theo dõi sự cải thiện về độ chính xác và giảm thiểu overfitting.

2.4.3. Kết quả

Kết quả huấn luyện và đánh giá mô hình được theo dõi qua các chỉ số như Precision, Recall, và mAP (mean Average Precision), giúp đánh giá hiệu quả nhận diện và phân đoạn. Trong nghiên cứu này, nhóm sử dụng hai kiến trúc YOLOv8-seg và YOLOv11-seg. YOLOv8-seg nổi bật với khả năng đa nhiệm (detection, segmentation, classification) và tốc độ xử lý cao, trong khi YOLOv11-seg được cải tiến về backbone và head, giúp tăng độ chính xác và ổn định hơn trong bài toán phân đoạn. Nhờ đó, nhóm có

thể so sánh trực tiếp khả năng nhận diện vật cản của từng mô hình để lựa chọn giải pháp tối ưu nhất.

Thực hiện huấn luyện mô hình với hai phiên bản YOLO, thu được bản so sánh và các kết quả:

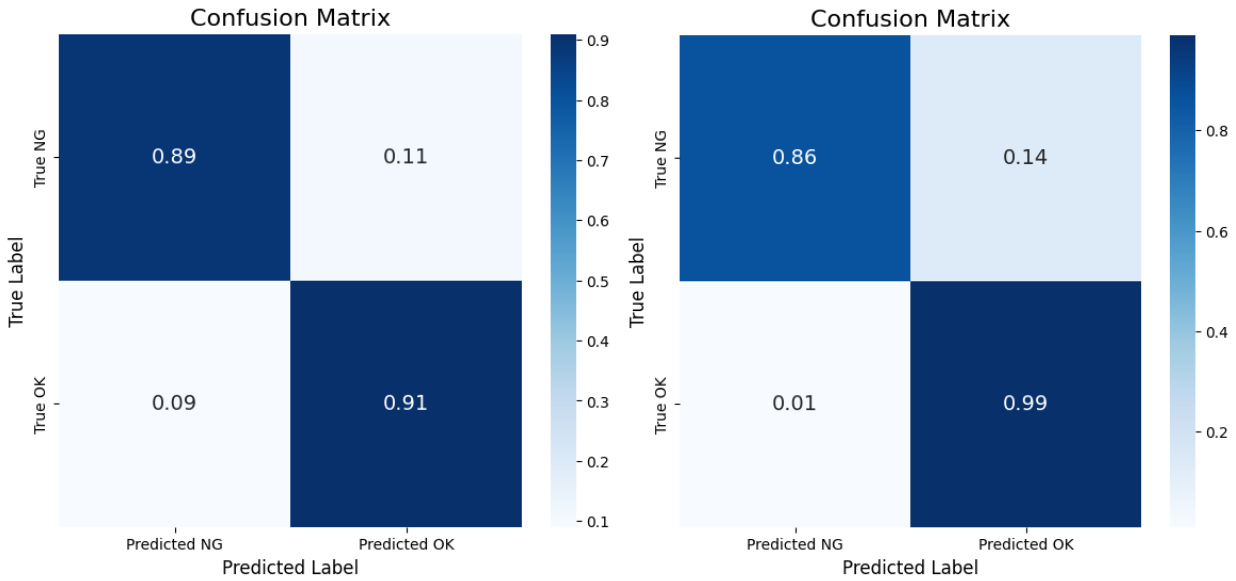
Bảng 2.3. *Bảng so sánh kết quả huấn luyện hai mô hình YOLO*

Mô hình	Thời gian huấn luyện	Tốc độ xử lý	mAP	Precision	Recall
YOLOv8-seg	0.637 giờ	7.2 ms	0.952	0.927	0.913
YOLOv11-seg	0.670 giờ	8.6 ms	0.972	0.894	0.940

Nhìn vào bảng so sánh kết quả huấn luyện hai mô hình YOLO trên (Bảng 2.3), ta có thể nhận thấy rằng:

- Tốc độ huấn luyện và độ chính xác tổng thể (mAP) của YOLO11n-seg đều vượt trội hơn một xíu so với YOLOv8n-seg. Điều này phản ánh khả năng tối ưu hóa trong quá trình huấn luyện, giúp mô hình vừa đạt tốc độ nhanh hơn vừa duy trì độ chính xác tổng thể cao hơn.

- Xét về Precision và Recall, hai mô hình thể hiện đặc điểm bổ sung cho nhau. YOLOv8n-seg đạt Precision cao hơn (0.927), đồng nghĩa với việc giảm thiểu số lượng dự đoán sai, nhưng lại có Recall thấp hơn (0.913) nên dễ bỏ sót các mẫu NG. Ngược lại, YOLO11n-seg đạt Recall cao hơn (0.940), cho thấy khả năng phát hiện mẫu NG tốt hơn, song Precision thấp hơn khiến vẫn tồn tại một số dự đoán sai. Ta có thể nhìn thấy rõ ở kết quả ma trận nhầm lẫn của 2 mô hình qua Hình 2.20 dưới đây:



Hình 2.20. *Ma trận nhầm lẫn của 2 mô hình YOLO*

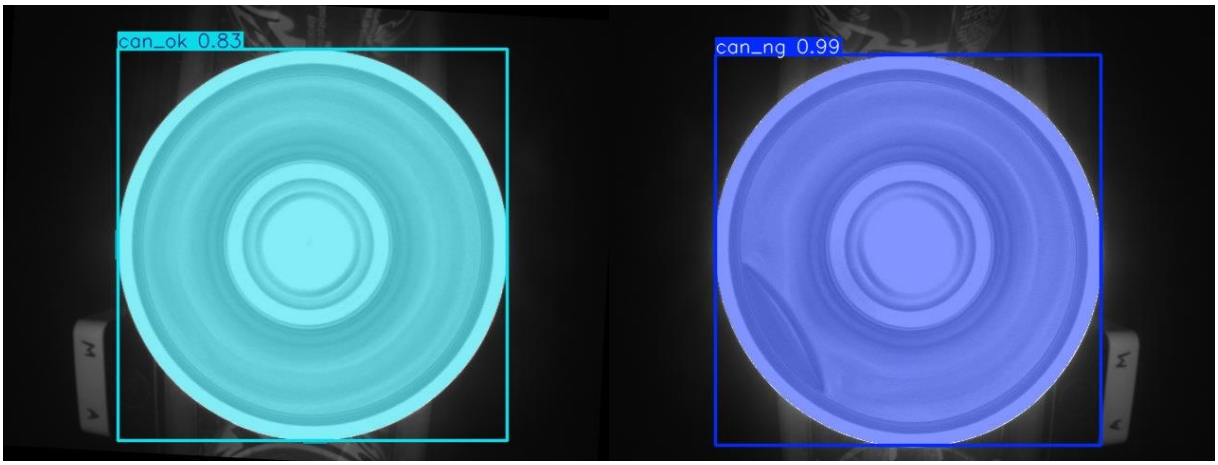
(Bên phải: YOLOv8-seg, bên trái: YOLOv11-seg)

Trong môi trường sản xuất công nghiệp, yêu cầu quan trọng nhất là phát hiện chính xác lon lỗi (NG) để loại bỏ khỏi dây chuyền, đồng thời duy trì tốc độ xử lý cao nhằm không làm gián đoạn sản xuất.

- YOLO11n-seg với Recall cao (0.94) và mAP đạt 0.972 cho thấy khả năng phát hiện lỗi NG gần như toàn diện. Điều này đặc biệt hữu ích trong thực tế, vì bỏ sót một lon lỗi có thể gây ảnh hưởng đến chất lượng sản phẩm và uy tín thương hiệu. Tuy nhiên, Precision thấp hơn nghĩa là vẫn có một số lon đạt bị nhầm thành lỗi, dẫn đến loại bỏ thừa.

- YOLOv8n-seg lại có Precision cao hơn (0.927) và tốc độ xử lý nhanh hơn. Điều này giúp giảm thiểu việc loại bỏ nhầm lon đạt, đồng thời phù hợp với yêu cầu thời gian thực trên dây chuyền tốc độ cao. Tuy nhiên, Recall thấp hơn (0.913) khiến nguy cơ bỏ sót một số lon lỗi.

Kết quả thực nghiệm mô hình trên một vài ảnh thực tế thu được như sau:



Hình 2.21. *Hình minh hoạ kết quả dự đoán mô hình YOLO*

2.5. Kiểm thử và đánh giá mô hình

2.5.1. Tập dữ liệu kiểm thử

Tập dữ liệu kiểm thử DATA_TEST bao gồm tổng cộng 1000 ảnh được phân bổ cân bằng (500 ảnh OK và 500 ảnh NG) và được lấy từ 2 nguồn dữ liệu giúp đảm bảo tính đại diện và cân bằng giữa các lớp, đồng thời phản ánh rõ hơn hiệu năng của mô hình trong nhiều tình huống khác nhau. Tập dữ liệu được mô tả chi tiết tại Bảng 2.4 dưới đây.

Bảng 2.4. *Mô tả tập dữ liệu kiểm thử*

Nguồn dữ liệu	Lon đạt chuẩn	Lon không đạt chuẩn	Tổng cộng	Mục đích
Từ dữ liệu huấn luyện	250 ảnh	250 ảnh	500 ảnh	Kiểm tra khả năng ghi nhớ của mô hình trên dữ liệu từng xuất hiện trong tập huấn luyện.
Dữ liệu mới	250 ảnh	250 ảnh	500 ảnh	Dùng để đánh giá khả năng tổng quát hóa và nhận diện các lỗi chưa từng thấy.
Tổng cộng	500 ảnh	500 ảnh	1000 ảnh	Đảm bảo tính cân bằng giữa các lớp

2.5.2. Phương pháp kiểm thử

Sử dụng mô hình đã huấn luyện tốt nhất (best.pt) của từng mô hình để chạy suy luận (inference) trên toàn bộ 1.000 ảnh của tập dữ liệu kiểm thử mới này. Kết quả đầu ra được đánh giá theo chỉ số:

Độ chính xác cấp độ ảnh (Image-level Accuracy): Tính tỷ lệ phần trăm ảnh được phân loại đúng (True Class ID được phát hiện) trên tổng số ảnh trong tập kiểm thử.

Công thức:

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{Số\ ảnh\ dự\ đoán\ đúng\ lớp}}{\boldsymbol{Tổng\ số\ ảnh\ cùng\ lớp}} \times 100\%. \quad (1.10)$$

2.5.3. Kết quả kiểm thử

Bảng 2.5. *Bảng kết quả kiểm thử*

Mô hình	Accuracy OK (%)	Accuracy NG (%)	Average Accuracy (%)
EfficientNet	92.4	71.6	82.0
Autoencoder	85.0	68.8	76.9
YOLOv8-seg	99.8	79.0	89.4
YOLOv11-seg	94.6	60.2	77.4

Nhận xét:

- YOLOv8-seg đạt độ chính xác cao nhất ở cả lớp OK (99.8%) và NG (79.0%), cho thấy khả năng phân đoạn và phân loại vượt trội, đặc biệt trong môi trường kiểm tra lon rỗng.

- EfficientNet có độ chính xác OK cao (92.4%) nhưng NG thấp hơn (71.6%), phù hợp với bài toán phân loại nhưng không tối ưu cho phát hiện lỗi NG.

- Autoencoder có độ chính xác thấp nhất ở lớp NG (68.8%), do đặc thù mô hình không huấn luyện trên dữ liệu lỗi, chỉ phát hiện bất thường qua lỗi tái tạo.

- YOLOv11-seg gây bất ngờ khi có độ chính xác NG thấp nhất (60.2%) dù là phiên bản mới hơn, cho thấy mô hình có thể chưa tối ưu cho bài toán cụ thể này hoặc cần tinh chỉnh thêm.

PHẦN V. KẾT LUẬN VÀ KIẾN NGHỊ

1. Kết luận

Đề tài đã thực hiện thành công việc khảo sát, triển khai và đánh giá bốn mô hình học sâu đại diện cho ba hướng tiếp cận chính trong bài toán kiểm tra chất lượng lon rỗng: phát hiện bất thường (Autoencoder), phân loại hình ảnh (EfficientNet), và phát hiện – phân đoạn đối tượng (YOLOv8-seg, YOLOv11-seg) trên tập dữ liệu thực tế thu thập từ dây chuyền sản xuất bia tại Carlsberg Việt Nam. Kết quả kiểm thử cho thấy:

- YOLOv8-seg là mô hình hiệu quả nhất, đạt độ chính xác trung bình 89.4%, đặc biệt nổi bật ở lớp OK (99.8%) và NG (79.0%), cho thấy khả năng phân đoạn và nhận diện lỗi vượt trội. Đây là mô hình khả thi nhất để áp dụng vào thực tế kiểm tra lon rỗng.

- EfficientNet và Autoencoder cho thấy tiềm năng ở các hướng tiếp cận khác nhau, nhưng độ chính xác lớp NG còn hạn chế, khó đáp ứng yêu cầu công nghiệp nếu dùng độc lập. Autoencoder thể hiện khả năng phát hiện bất thường trong điều kiện thiếu dữ liệu NG, nhưng độ chính xác thấp hơn (76.9%) do đặc thù không giám sát.

- YOLOv11-seg, dù là phiên bản mới nhưng lại cho kết quả chưa như kỳ vọng (77.4%), đặc biệt ở lớp NG (60.2%), cho thấy cần tinh chỉnh thêm để phù hợp với bài toán cụ thể.

Từ kết quả này, có thể khẳng định rằng việc ứng dụng học sâu trong kiểm tra chất lượng lon rỗng là hoàn toàn khả thi, tuy nhiên độ chính xác đối với lon NG vẫn chưa tối ưu. Đây là thách thức thực tế bởi dữ liệu NG hiếm, đa dạng hình thái và khó thu thập.

2. Hạn chế

Mặc dù đề tài đã đạt được những kết quả tích cực, vẫn tồn tại một số hạn chế cần được nhìn nhận rõ ràng:

- Hạn chế về dữ liệu NG (lon lỗi): Số lượng lon NG trong thực tế rất ít, do nhà cung cấp đã kiểm định chất lượng trước khi đưa vào dây chuyền. Điều này khiến tập dữ

liệu NG không đủ phong phú để mô hình học được đầy đủ các dạng lỗi. Các lỗi NG thường hiếm, đa dạng về hình thái (móp nhẹ, mép biến dạng, vết lõm nhỏ, ánh sáng phản xạ bất thường,..), nên mô hình dễ bị thiên lệch về lớp OK. Đây là nguyên nhân chính khiến độ chính xác của lớp NG chưa cao.

- Độ khó của dữ liệu thực tế: Ảnh chụp lon rỗng trong dây chuyền thường chịu ảnh hưởng bởi ánh sáng không ổn định, nhiễu nền, góc chụp thay đổi. Những yếu tố này làm cho lỗi NG càng khó nhận diện, đặc biệt khi lỗi rất nhỏ hoặc bị che khuất. Mô hình học sâu vốn nhạy cảm với dữ liệu đầu vào, nên nếu dữ liệu không được chuẩn hóa tốt, kết quả sẽ dao động.

3. Kiến nghị và hướng phát triển

Để nâng cao hiệu quả và tiến tới áp dụng thực tế trong dây chuyền sản xuất, đề tài đưa ra một số kiến nghị và định hướng phát triển sau:

- Mở rộng dữ liệu NG: Đây là yếu tố quan trọng nhất. Cần tiếp tục thu thập thêm dữ liệu lon lỗi từ nhiều ca sản xuất khác nhau, nhiều điều kiện ánh sáng và góc chụp. Ngoài ra, có thể áp dụng các kỹ thuật data augmentation (xoay, thay đổi độ sáng, thêm nhiễu) hoặc tổng hợp dữ liệu giả lập (synthetic defects) bằng GAN/U-Net để mô phỏng các dạng lỗi hiếm. Việc này giúp mô hình học được nhiều đặc trưng NG hơn, giảm tình trạng bỏ sót lỗi.

- Kết hợp nhiều phương pháp: xây dựng hệ thống kiểm tra đa tầng, ví dụ YOLOv8-seg làm mô hình chính, EfficientNet hoặc Autoencoder làm lớp xác nhận bổ sung để giảm sai sót.

- Tinh chỉnh mô hình mới: thử nghiệm các thuật toán tối ưu hóa (Lion Optimizer, AdamW), Attention Mechanisms hoặc kiến trúc lai để cải thiện khả năng nhận diện NG.

- Thử nghiệm thực tế trên dây chuyền: đánh giá tốc độ suy luận, độ ổn định, khả năng tích hợp với robot gắp và băng chuyền, từ đó điều chỉnh mô hình cho phù hợp với yêu cầu sản xuất.

- Phát triển hệ thống giám sát thông minh: kết hợp IoT và thị giác máy tính để theo dõi chất lượng lon rỗng theo thời gian thực, cảnh báo sớm khi phát hiện lỗi.

Đề tài đã chứng minh tính khả thi của việc ứng dụng học sâu trong kiểm tra lon rỗng. Tuy nhiên, để đạt độ chính xác tối ưu cho lon không đạt chuẩn NG và đảm bảo tính ứng dụng trong công nghiệp, cần tăng cường dữ liệu, kết hợp nhiều phương pháp, và thử nghiệm thực tế. Đây sẽ là nền tảng quan trọng để phát triển hệ thống kiểm tra chất lượng lon rỗng tự động, góp phần nâng cao hiệu quả và độ tin cậy của dây chuyền sản xuất bia trong bối cảnh Cách mạng công nghiệp 4.0.

TÀI LIỆU THAM KHẢO

[1] C. Das, A. K. Sahoo, and C. Pradhan, "Multicriteria recommender system using different approaches," in *Cognitive Big Data Intelligence with a Metaheuristic Approach*, S. Mishra, H. K. Tripathy, P. K. Mallick, A. K. Sangaiah, and G.-S. Chae, Eds., Academic Press, pp. 259-277, 2022.

[2] D. Kang, J. Lee, and H. Kim, "Surface defect detection of metals using deep learning-based computer vision," *Robotics and Computer-Integrated Manufacturing*, vol. 59, p. 101848, 2019.

[3] N. De Silva, "Mechanical defects in aluminum can bodies and their impact on filling line stability," *Journal of Materials Processing Technology*, vol. 285, p. 116512, 2020.

[4] G. Pang, C. Shen, L. Cao, and A. van den Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surveys*, vol. 54, no. 4, pp. 1–38, Apr. 2021, doi: 10.1145/3439950.

[5] Canadian Food Inspection Agency, "Metal can defects: Identification and classification," Government of Canada, Ottawa, ON, Canada, 2017. Accessed: Dec. 9, 2025. [Online]. Available: https://inspection.canada.ca/en/preventive-controls/controls-food/metal-can-defects

[6] Z. Li, Y. Yan, X. Wang, Y. Ge, and L. Meng, "A survey of deep learning for industrial visual anomaly detection," *Artificial Intelligence Review*., vol. 58, no. 9, p. 279, 2025.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[10] M. K. Shereen, M. I. Khattak, and M. Al-Hasan, "A frequency and radiation pattern combo-reconfigurable novel antenna for 5G applications and beyond," *Electronics*, vol. 9, no. 9, p. 1372, Aug. 2020, doi: 10.3390/electronics9091372.

[11] Z. Li, Y. Yan, X. Wang, Y. Ge, and L. Meng, "A survey of deep learning for industrial visual anomaly detection," Artificial Intelligence Review, vol. 58, no. 9, p. 279, 2025.

[12] Y. Zhang, H. Liu, and J. Wang, "Real-Time Plastic Surface Defect Detection Using Deep Learning," *IEEE Access*, vol. 10, pp. 57912–57923, 2022. doi: 10.1109/access.2022.9794475

[13] E. Gustafsson, S. Thomee, A. Grimby-Ekman, and M. Hagberg, "Texting on mobile phones and musculoskeletal disorders in young adults: A five-year cohort study," *Applied Ergonomics*, vol. 58, pp. 208-214, 2017. doi: 10.1016/j.apergo.2017.04.012

[14] Z. Zhang, G. Fu, R. Ni, J. Liu, và X. Yang, "A generative method for steganography by cover synthesis with auxiliary semantics," *Tsinghua Science and Technology*, vol. 25, no. 4, pp. 516-527, 2020.

[15] T. M. Mitchell, "Does machine learning really work?," *AI Magazine*, vol. 18, no. 3, p. 11, 1997.

[16] E. Alpaydin, *Introduction to Machine Learning*, Cambridge, MA, USA: MIT Press, 2020.

[17] T. Hastie, R. Tibshirani, J. Friedman, và J. Franklin, "The elements of statistical learning: data mining, inference and prediction," *The Mathematical Intelligencer*, vol. 27, no. 2, pp. 83-85, 2005.

[18] A. Krizhevsky, I. Sutskever, và G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[19] A. Krizhevsky, I. Sutskever, và G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, 2017.

[20] J. B. MacQueen, “Some Methods for Classification and Analysis of Multivariate Observations,” in *Proc. of the 5th Berkeley Symp. on Mathematical Statistics and Probability*, pp. 281–297, 1967.

[21] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006, doi: 10.1126/science.1127647.

[22] X. Zhu and A. B. Goldberg, "Introduction to Semi-Supervised Learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 3, no. 1, pp. 1–130, 2009, doi: 10.2200/S00196ED1V01Y200906AIM006.

[23] O. Chapelle, B. Schölkopf, và A. Zien, “A discussion of semi-supervised learning and transduction,” *In Semi-supervised learning*, MIT Press, pp. 473-478, 2006.

[24] G. E. Hinton, S. Osindero, và Y. W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, no. 7, pp. 1527-1554, 2006.

[25] Y. LeCun, Y. Bengio, và G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.

[26] Y. Bengio, Y. Lecun, và G. Hinton, “Deep learning for AI,” *Communications of the ACM*, vol. 64, no. 7, pp. 58-65, 2021.

[27] P. D. Khanh, "Convolutional Neural Network," phamdinhkhanh.github.io, Aug. 22, 2019. [Online]. Available: https://phamdinhkhanh.github.io [Accessed: Nov. 50, 2025].

[28] W. El-Shafai, N. El-Hag, A. Sedik, G. Elbanby, F. Abd El-Samie, N. F. Soliman, ..., and M. E. Abdel Samea, "An efficient medical image deep fusion model based on convolutional neural networks," Comput. Mater. Contin, vol. 74, no. 2, pp. 2905-2925, 2023.

[29] CS231n, "Convolutional Neural Networks for Visual Recognition," [Online]. Available: https://cs231n.github.io/convolutional-networks/. [Accessed: Nov. 20, 2025].

[30] Yu, D., Wang, H., Chen, P., and Wei, Z., "Mixed pooling for convolutional neural networks," in *International Conference on Rough Sets and Knowledge Technology*, Cham: Springer International Publishing, pp. 364-375, Oct. 2014.

[31] Xu, Q., Zhang, M., Gu, Z., and Pan, G., "Overfitting remedy by sparsifying regularization on fully-connected layers of CNNs," Neurocomputing, vol. 328, pp. 69-74, 2019.

[32] "Introduction to Autoencoders: From The Basics to Advanced Applications in PyTorch," DataCamp, updated Dec. 14, 2023. [Online]. Available: https://www.datacamp.com/tutorial/introduction-to-autoencoders

[33] Ibomoiye Domor Mienye and Theo G. Swart, "Deep Autoencoder Neural Networks: A Comprehensive Review and New Perspectives," *Archives of Computational Methods in Engineering,* vol. 32, pp. 3981-4000, 2025.

[34] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, PMLR, pp. 6105-6114, May 2019.

[35] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779-788, 2016.

[36] Ultralytics, "Ultralytics YOLO," GitHub repository, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics. [Accessed: Nov. 20, 2025].

[37] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779-788, 2016.

[38] Python Software Foundation, "Welcome to Python.org," Python.org, 2025. [Online]. Available: https://www.python.org/. [Accessed: Dec. 5, 2025].

[39] Stanford University, "python-review.pdf," 2025. [Online]. Available: https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184/lectures/python-review.pdf. [Accessed: Dec. 5, 2025].

[40] Django Software Foundation, "Django," Django Project, 2025. [Online]. Available: https://www.djangoproject.com/. [Accessed: Dec. 5, 2025].

[41] Pallets Projects, "Welcome to Flask-Flask Documentation (3.1.x)," 2025. [Online]. Available: https://flask.palletsprojects.com/en/stable/. [Accessed: Dec. 5, 2025].

[42] NumPy Developers, "NumPy -," 2025. [Online]. Available: https://numpy.org/. [Accessed: Dec. 5, 2025].

[43] The pandas development team, "pandas - Python Data Analysis Library," 2025. [Online]. Available: https://pandas.pydata.org/. [Accessed: Dec. 5, 2025].

[44] scikit-learn developers, "scikit-learn: machine learning in Python — scikit-learn 1.5.2 documentation," 2025. [Online]. Available: https://scikit-learn.org/stable/. [Accessed: Dec. 5, 2025].

[45] TensorFlow Authors, "TensorFlow," TensorFlow, 2025. [Online]. Available: https://www.tensorflow.org/?hl=vi. [Accessed: Dec. 5, 2025].

[46] PyTorch Contributors, "PyTorch," 2025. [Online]. Available: https://pytorch.org/. [Accessed: Dec. 5, 2025].

[47] Sunsetting Python 2," Python.org, 2024. [Online]. Available: https://www.python.org/doc/sunset-python-2/. [Accessed: Dec. 5, 2025].

[48] "Python Documentation," Python.org, 2024. [Online]. Available: https://docs.python.org/3/. [Accessed: Dec. 5, 2025].

[49] "Roboflow: Computer vision tools for developers and enterprises," Roboflow, 2024. [Online]. Available: https://roboflow.com/. [Accessed: Dec. 5, 2025].

[50] "Google Colab," Google, 2024. [Online]. Available: https://colab.research.google.com/. [Accessed: Dec. 5, 2025].

[51] Ultralytics, "Ultralytics YOLO11," GitHub repository, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics. [Accessed: Dec. 5, 2025].

[52] BigDataUni, "Phương pháp đánh giá mô hình phân loại (Classification Model Evaluation)," BigDataUni, 2025. [Online]. Available: https://bigdatauni.com/tin-tuc/phuong-phap-danh-gia-mo-hinh-phan-loai-classification-model-evalutation.html. [Accessed: Dec. 5, 2025].

[53] Das, A. K. Sahoo, và C. Pradhan, "Multicriteria recommender system using different approaches," trong *Cognitive Big Data Intelligence with a Metaheuristic Approach*, S. Mishra, H. K. Tripathy, P. K. Mallick, A. K. Sangaiah, và G.-S. Chae, biên tập, Academic Press, 2022, tr. 259-277.

[54] K. M. Ting, "Precision and recall," in *Encyclopedia of Machine Learning*, 2011, pp. 781, doi: 10.1007/978-0-387-30164-8_652.

[55] "Phương pháp đánh giá mô hình phân loại (Classification Model Evaluation)," BigDataUni. [Online]. Available: https://bigdatauni.com/tin-tuc/phuong-phap-danh-gia-mo-hinh-phan-loai-classification-model-evalutation.html. [Accessed: Dec. 5, 2025].

[56] K. Chumachenko, M. Gabbouj, and A. Iosifidis, “Chapter 11 - Object detection and tracking,” in *Deep Learning for Robot Perception and Cognition*, A. Iosifidis and A. Tefas, Eds., Academic Press, 2022, pp. 243–278. doi: https://doi.org/10.1016/B978-0-32-385787-1.00016-6.