

**BỘ CÔNG THƯƠNG**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**



**ĐỒ ÁN TỐT NGHIỆP**  
**NGÀNH KHOA HỌC MÁY TÍNH**

**ĐỀ TÀI: XÂY DỰNG HỆ THỐNG ƯỚC LƯỢNG**  
**CALO TRONG BỮA ĂN BẰNG MÔ HÌNH YOLOv8**

<b>Giảng viên hướng dẫn:</b>	TS. Nguyễn Mạnh Cường
<b>Sinh viên thực hiện:</b>	Nguyễn Hồng Quân
<b>Mã sinh viên:</b>	2021600807

*Hà Nội, 2025*

## LỜI CẢM ƠN

Đầu tiên, em xin gửi lời cảm ơn, lời tri ân chân thành nhất đến các thầy cô trường Đại học Công Nghiệp Hà Nội, đặc biệt là các thầy cô trường Công nghệ Thông tin và Truyền thông đã tận tình giảng dạy, truyền đạt kiến thức quý báu cho em trong suốt quá trình học tập, và đã tạo điều kiện cho em thực hiện đồ án tốt nghiệp. Bên cạnh đó, em cũng muốn gửi lời cảm ơn tới những người bạn đã quan tâm và giúp đỡ em trong suốt thời gian qua. Hơn hết, em xin bày tỏ lòng biết ơn sâu sắc đến Tiến sĩ Nguyễn Mạnh Cường, người đã tận tâm hướng dẫn, hỗ trợ và đồng hành cùng em trong suốt quá trình thực hiện đề tài.

Trong quá trình thực hiện đồ án và viết báo cáo, em khó tránh khỏi những thiếu sót. Em mong nhận được sự góp ý từ thầy cô để có thể tiếp tục học hỏi, nâng cao kiến thức và hoàn thiện tốt hơn bài báo cáo.

Em xin trân trọng cảm ơn!

Sinh viên thực hiện

*Nguyễn Hồng Quân*

## MỤC LỤC

<b>LỜI CẢM ƠN .....</b>	<b>i</b>
<b>DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT .....</b>	<b>iv</b>
<b>DANH MỤC BẢNG .....</b>	<b>v</b>
<b>DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ .....</b>	<b>vi</b>
<b>MỞ ĐẦU .....</b>	<b>viii</b>
<b>CHƯƠNG 1. TỔNG QUAN VỀ BÀI TOÁN ƯỚC LƯỢNG CALO DỰA TRÊN HÌNH ẢNH VÀ PHÁT BIỂU BÀI TOÁN.....</b>	<b>1</b>
1.1. Tổng quan về dinh dưỡng và vấn đề theo dõi calo.....	1
1.1.1. Calo .....	1
1.1.2. Vấn đề theo dõi calo.....	1
1.1.3. Khó khăn trong việc tính toán calo thủ công .....	2
1.1.4. Giải pháp .....	3
1.2. Cơ sở lý thuyết.....	4
1.2.1. Một số khái niệm cơ bản .....	4
1.2.2. Giới thiệu về mạng nơ-ron tích chập.....	5
1.2.3. Giới thiệu về phát hiện đối tượng .....	6
1.3. Phát biểu bài toán.....	8
1.3.1. Giới thiệu bài toán .....	8
1.3.2. Phát biểu bài toán .....	8
1.3.3. Các khó khăn và thách thức của bài toán .....	9
1.3.4. Các miền ứng dụng của bài toán .....	10
<b>CHƯƠNG 2. MỘT SỐ KỸ THUẬT GIẢI QUYẾT BÀI TOÁN.....</b>	<b>11</b>
2.1. Phương hướng tiếp cận bài toán .....	11
2.2. Một số kỹ thuật giải quyết bài toán .....	12
2.2.1. Support Vector Machine .....	12
2.2.2. Mạng nơ-ron tích chập .....	15
2.2.3. YOLOv8.....	18
2.3. Hồi quy tuyến tính .....	27
<b>CHƯƠNG 3. THỰC NGHIỆM .....</b>	<b>30</b>

3.1. Quy trình thực nghiệm.....	30
3.2. Dữ liệu thực nghiệm .....	31
3.3. Tiền xử lý dữ liệu.....	35
3.4. Huấn luyện mô hình.....	37
3.4.1. YOLOv8.....	37
3.4.2. Hồi quy tuyến tính.....	39
3.5. Kết quả thực nghiệm.....	40
3.5.1. YOLOv8.....	40
3.5.2. Hồi quy tuyến tính.....	45
<b>CHƯƠNG 4. XÂY DỰNG HỆ THỐNG.....</b>	<b>47</b>
4.1. Giới thiệu một số công nghệ sử dụng.....	47
4.1.1. Web framework.....	47
4.1.2. Hệ quản trị cơ sở dữ liệu .....	48
4.2. Phân tích thiết kế hệ thống.....	49
4.2.1. Biểu đồ use case .....	49
4.2.2. Đặc tả use case .....	50
4.3. Một số kết quả .....	54
<b>KẾT LUẬN.....</b>	<b>60</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>62</b>

**DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT**

<b>Viết tắt</b>	<b>Ý nghĩa</b>
AI	Artificial Intelligence
YOLO	You Only Look Once
CNN	Convolutional Neural Network
SVM	Support Vector Machine
AP	Average Precision
mAP	mean Average Precision
IoU	Intersection over Union
NMS	Non-Maximum Suppression

## DANH MỤC BẢNG

Bảng 2.1 Bảng thông tin hiệu suất của các phiên bản YOLOv8 trên tập dữ liệu COCO .....	26
Bảng 3.1 Bảng hiệu suất sau khi huấn luyện mô hình .....	44
Bảng 4.1 Bảng mô tả use case Đăng ký .....	50
Bảng 4.2 Bảng mô tả use case Đăng nhập .....	51
Bảng 4.3 Bảng mô tả use case Xem thông tin .....	52
Bảng 4.4 Bảng mô tả use case Thay đổi thông tin .....	53

## DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

Hình 1.1	Mối quan hệ giữa trí tuệ nhân tạo, học máy và học sâu [4].....	4
Hình 1.2	Kiến trúc CNN [6].....	5
Hình 1.3	Hình ảnh ví dụ về nhận dạng đối tượng trong ảnh [7].....	6
Hình 1.4	Phân loại các thuật toán phát hiện đối tượng [8] .....	7
Hình 2.1	Minh họa thuật toán SVM [10] .....	13
Hình 2.2	Kiến trúc CNN [12].....	16
Hình 2.3	Mốc thời gian các phiên bản của YOLO tính tới năm 2023 [9] .....	19
Hình 2.4	Kiến trúc CNN tạo thành xương sống của mô hình YOLO [9].....	20
Hình 2.5	Công thức và ví dụ về IoU [13] .....	22
Hình 2.6	Kết quả sau khi áp dụng kỹ thuật NMS [13] .....	24
Hình 2.7	Kiến trúc YOLOv8 [13].....	25
Hình 2.8	Minh họa thuật toán hồi quy tuyến tính [14] .....	27
Hình 3.1	Quy trình thực nghiệm .....	30
Hình 3.2	Một số hình ảnh trong bộ dữ liệu.....	32
Hình 3.3	Minh họa hai góc nhìn trong bộ dữ liệu.....	33
Hình 3.4	Kích thước vật tham chiếu đồng xu .....	33
Hình 3.5	Kích thước vật tham chiếu đĩa đỏ và đĩa trắng .....	34
Hình 3.6	Hình minh họa dữ liệu có trong trang tính “apple” .....	34
Hình 3.7	Số lượng ảnh trong mỗi tập.....	35
Hình 3.8	Kết quả tệp văn bản sau khi chuyển đổi tệp XML tương ứng.....	36
Hình 3.9	Thông tin cấu hình tệp data.yaml.....	36
Hình 3.10	Lệnh huấn luyện mô hình yolov8n .....	37
Hình 3.11	Kết quả huấn luyện mô hình YOLOv8 qua một vài epoch .....	39
Hình 3.12	Hàm xử lý dữ liệu và huấn luyện mô hình hồi quy tuyến tính .....	40
Hình 3.13	Hàm ghi các hệ số hồi quy ra tệp.....	40
Hình 3.14	Biểu đồ Precision-Confidence Curve.....	41
Hình 3.15	Biểu đồ Precision-Recall Curve.....	42
Hình 3.16	Biểu đồ Recall-Confidence Curve .....	43

Hình 3.17 Kết quả của mô hình .....	43
Hình 3.18 Kết quả huấn luyện mô hình hồi quy tuyến tính.....	46
Hình 4.1 Biểu đồ use case tổng quát.....	49
Hình 4.2 Biểu đồ phân rã use case Xem thông tin.....	50
Hình 4.3 Giao diện trang chủ khi người dùng truy cập .....	54
Hình 4.4 Giao diện màn hình đăng nhập .....	55
Hình 4.5 Giao diện màn hình đăng ký .....	55
Hình 4.6 Giao diện người dùng sau khi đã đăng nhập.....	56
Hình 4.7 Giao diện tải hình ảnh lên .....	56
Hình 4.8 Giao diện màn hình ước lượng calo.....	57
Hình 4.9 Giao diện xem thông tin.....	58
Hình 4.10 Giao diện thay đổi thông tin.....	58
Hình 4.11 Giao diện thông báo thành công thay đổi thông tin.....	59



## MỞ ĐẦU

Trong thời kỳ công nghệ phát triển mạnh mẽ và đầy vũ bão, AI ngày càng được ứng dụng trong nhiều ngành nghề, từ y tế, giáo dục cho đến thương mại điện tử. Vai trò của AI ngày một rõ ràng và quan trọng hơn trong nhiều lĩnh vực. Một ứng dụng nổi trội của AI là nhận diện hình ảnh, cho phép tự động hóa những nhiệm vụ phức tạp mà trước kia cần đến sự tham gia của con người.

Hiện nay, mọi người đã chú trọng đến sức khỏe bản thân, đặc biệt là chế độ dinh dưỡng và quản lý lượng calo tiêu thụ mỗi bữa ăn. Các hệ thống tự động nhận diện thực phẩm và ước lượng lượng calo đã trở thành một lĩnh vực nghiên cứu giá trị. Việc áp dụng AI vào ước tính calo không chỉ hỗ trợ người dùng duy trì chế độ ăn uống lành mạnh mà còn góp phần theo dõi sức khỏe một cách hiệu quả hơn. Do đó, em đã quyết định lựa chọn đề tài: “Xây dựng hệ thống ước lượng calo trong bữa ăn bằng mô hình YOLOv8” nhằm hỗ trợ mọi người có thể theo dõi lượng calo hấp thụ vào trong mỗi bữa ăn để có thể theo dõi và cải thiện sức khỏe.

Trong quá trình thực hiện đề tài, các mô hình học sâu như CNN và đặc biệt là mô hình phát hiện vật thể YOLO đã được triển khai để nhận dạng và phân đoạn chính xác các món ăn trong ảnh. Nhờ khả năng kết hợp giữa phát hiện đối tượng và phân đoạn ảnh, hệ thống có thể xác định vùng chiếm diện tích thực tế của từng món ăn, từ đó phục vụ cho việc tính toán và ước lượng calo chính xác hơn thông qua hình ảnh. Hệ thống ứng dụng mô hình YOLOv8, một trong những phiên bản tiên tiến nhất của YOLO, giúp nhận diện nhanh chóng và chính xác các đối tượng.

Mục tiêu chính của đề tài là xây dựng một hệ thống hỗ trợ người dùng trong việc kiểm soát lượng calo tiêu thụ, đặc biệt hữu ích cho những người có nhu cầu giảm cân, duy trì vóc dáng hoặc có các vấn đề sức khỏe cần theo dõi chế độ ăn uống. Ngoài ra, hệ thống có thể được mở rộng và tích hợp vào các

ứng dụng di động để hỗ trợ người dùng theo dõi lượng calo hàng ngày một cách tiện lợi.

Báo cáo được chia thành 4 chương như sau:

**Chương 1: Tổng quan về bài toán ước lượng calo dựa trên hình ảnh và phát biểu bài toán:** Nội dung của chương trình bày khái quát về vấn đề dinh dưỡng trong bữa ăn, các khái niệm về AI nói chung, lĩnh vực nhận dạng hình ảnh và mạng nơ-ron nói riêng, và phát biểu bài toán.

**Chương 2: Một số kỹ thuật giải quyết bài toán:** Chương này sẽ giới thiệu về các phương pháp nhận diện món ăn và tính toán, ước lượng calo trong bữa ăn cùng với ưu và nhược điểm của mỗi phương pháp, cùng với đó là đề xuất giải pháp của mình cho bài toán.

**Chương 3: Thực nghiệm:** Chương 3 chủ yếu sẽ trình bày chi tiết về quy trình thực nghiệm của báo cáo.

**Chương 4: Xây dựng hệ thống:** Nội dung của chương 4 là quá trình triển khai hệ thống và kết quả đạt được khi triển khai hệ thống.

Qua quá trình thực hiện đồ án tốt nghiệp, em đã có thêm cơ hội tìm hiểu sâu hơn về các kỹ thuật nhận diện hình ảnh, đặc biệt là ứng dụng mô hình YOLOv8 trong việc nhận diện thực phẩm và ước lượng calo. Em hy vọng rằng báo cáo này sẽ cung cấp một góc nhìn chi tiết về quá trình xây dựng hệ thống ước lượng calo từ hình ảnh, đồng thời có thể là tài liệu tham khảo hữu ích cho các nghiên cứu sau này.

# CHƯƠNG 1. TỔNG QUAN VỀ BÀI TOÁN ƯỚC LƯỢNG CALO DỰA TRÊN HÌNH ẢNH VÀ PHÁT BIỂU BÀI TOÁN

## 1.1. Tổng quan về dinh dưỡng và vấn đề theo dõi calo

### 1.1.1. Calo

Calo (calorie) là đơn vị đo năng lượng mà cơ thể hấp thụ từ thực phẩm. Mỗi hoạt động của cơ thể, từ hô hấp, tuần hoàn máu đến vận động hàng ngày, đều cần một lượng calo nhất định. Nếu calo nạp vào lớn hơn calo tiêu hao, cơ thể sẽ tích tụ năng lượng dưới dạng mỡ, gây tăng cân. Ngược lại, nếu lượng calo tiêu hao lớn hơn lượng nạp vào, cơ thể sẽ giảm cân [1].

### 1.1.2. Vấn đề theo dõi calo

*(Các số liệu trong mục 1.1.2 được trích dẫn từ tài liệu tham khảo [2])*

Trong những năm gần đây, người Việt Nam ngày càng quan tâm hơn đến chế độ ăn uống và sức khỏe. Điều này thể hiện rõ qua xu hướng lựa chọn thực phẩm sạch, hữu cơ, cũng như áp dụng các phương pháp ăn uống khoa học nhằm cải thiện chất lượng bữa ăn hằng ngày. Theo khảo sát, cứ gần 2 trên 3 người tiêu dùng sẵn sàng chi trả thêm 10% cho các thực phẩm tự nhiên và hữu cơ. Điều này cho thấy người Việt đang có xu hướng ưu tiên những thực phẩm không chỉ đảm bảo giá trị dinh dưỡng mà còn an toàn, thân thiện với môi trường.

Ngoài ra, nhận thức về sức khỏe đang ngày càng rõ nét hơn trong việc lựa chọn thực phẩm. 26% người tiêu dùng Việt Nam đã thực hiện chế độ ăn kiêng trong vòng 1 năm qua, trong đó 47% tăng cường tiêu thụ trái cây và rau xanh, 43% giảm ăn vặt, 41% bổ sung thêm chất xơ và 40% chú trọng đến vitamin cũng như khoáng chất. Đồng thời, người Việt cũng có xu hướng thay đổi cách chế biến món ăn theo hướng lành mạnh hơn, với 52% hạn chế món chiên rán, 47% bổ sung thêm rau xanh và chất xơ, cùng 41% cắt giảm lượng đường trong khẩu phần ăn.

Bên cạnh chế độ ăn uống khoa học, thực phẩm chức năng cũng trở thành một lựa chọn phổ biến để bổ sung dinh dưỡng và phòng ngừa bệnh tật. Theo khảo sát, 62% người dùng mong muốn tăng cường thể lực thông qua thực phẩm chức năng, 53% quan tâm đến việc cải thiện hệ miễn dịch, 44% tập trung vào sức khỏe xương khớp, 40% muốn hỗ trợ hệ tiêu hóa, và 38% tìm kiếm giải pháp giúp phát triển trí não. Đặc biệt, đối với phụ nữ, 47% mong muốn chống lão hóa, trong khi 48% kỳ vọng vào các sản phẩm làm đẹp da và tóc.

Một xu hướng khác cũng đang được người tiêu dùng Việt quan tâm là việc kiểm tra thông tin dinh dưỡng trên bao bì sản phẩm. Có đến 81% người muốn thấy thông tin dinh dưỡng rõ ràng, 80% mong muốn biết mức độ tự nhiên của sản phẩm, và 78% chú trọng đến các thành phần có lợi cho sức khỏe. Trên thực tế, 1 trên 3 người tiêu dùng có thói quen kiểm tra thông tin trước khi mua hàng, với 55% đọc bảng thành phần dinh dưỡng, 41% tìm kiếm biểu tượng chứng nhận sức khỏe, và 25% quan tâm đến nhãn thực phẩm hữu cơ.

Tuy nhiên, mặc dù đã có nhiều biện pháp để kiểm soát chế độ ăn uống, nhưng một vấn đề lớn vẫn tồn tại: làm thế nào để tính toán chính xác lượng calo tiêu thụ? Hiện nay, việc theo dõi lượng calo chủ yếu dựa vào việc nhập dữ liệu thủ công, khiến nhiều người gặp khó khăn trong quá trình kiểm soát khẩu phần ăn. Trong khi đó, những yếu tố quan trọng như hàm lượng chất béo, đường, protein, chất xơ và omega lại đóng vai trò quyết định đến sức khỏe tổng thể. Theo khảo sát, có 51% người tiêu dùng quan tâm đến lượng chất béo, 49% kiểm tra hàm lượng đường, 66% chú ý đến vitamin, 50% theo dõi lượng protein và 45% chú trọng đến chất xơ.

### **1.1.3. Khó khăn trong việc tính toán calo thủ công**

Trước khi các kỹ thuật học sâu bùng nổ, việc ước lượng lượng calo của bữa ăn thường được thực hiện bằng phương pháp đo lường thủ công. Quá trình này bao gồm một số bước chính sau:

- Xác định loại thực phẩm và khẩu phần ăn bằng cách nhận diện đầy đủ các thành phần có trong bữa ăn. Mỗi loại thực phẩm mang giá trị năng lượng khác nhau tùy thuộc vào thành phần dinh dưỡng. Cần tiến hành đo lường hoặc ước lượng chính xác khối lượng của từng thành phần, tính bằng gam.
- Tính toán lượng calo trong từng thành phần dựa trên bảng dữ liệu dinh dưỡng chuẩn. Cần lưu ý đến sự thay đổi giá trị calo do phương pháp chế biến như luộc, chiên, hấp, nướng, hoặc do việc thêm gia vị, dầu mỡ.
- Cộng tổng lượng calo từ tất cả các thành phần, món ăn để xác định tổng năng lượng mà khẩu phần ăn cung cấp. Kết quả cho phép chúng ta đánh giá mức năng lượng nạp vào cơ thể.

Việc tính toán lượng calo thủ công này có tính chính xác cao, tuy nhiên lại có một số hạn chế nhất định như:

- Gây bất tiện vì phải tính toán nhiều.
- Không áp dụng được trong các tình huống đời thực, khi nhiều người không biết lượng calo tương ứng với từng loại thực phẩm, món ăn.

#### **1.1.4. Giải pháp**

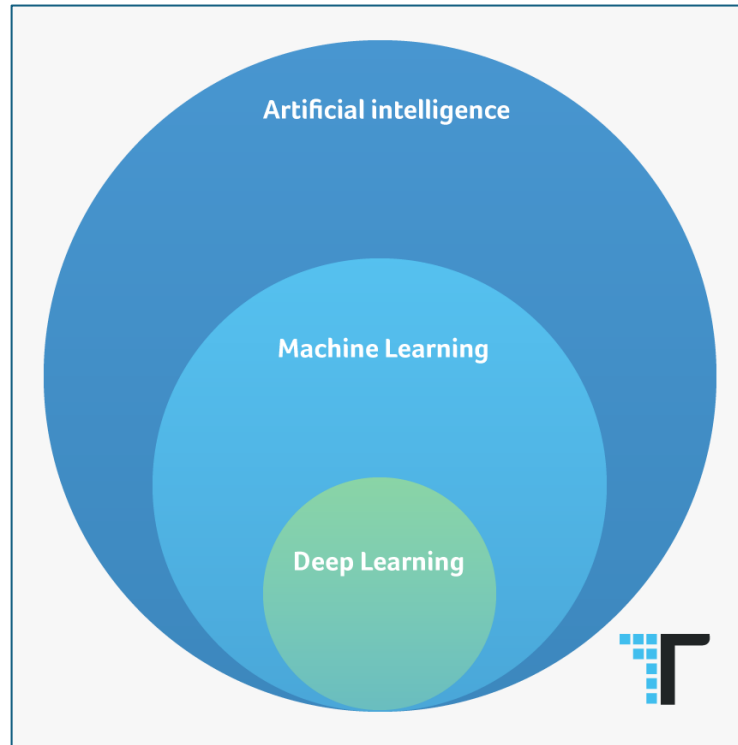
Thị giác máy tính và AI ra đời có thể giúp chúng ta theo dõi bữa ăn hàng ngày và thậm chí có thể là chuyên gia dinh dưỡng cá nhân của bạn. Những công nghệ này có thể được sử dụng để giúp cải thiện sức khỏe mỗi người bằng cách cung cấp theo dõi chính xác và kế hoạch ăn kiêng phù hợp, đồng thời giảm chi phí chăm sóc sức khỏe và giúp cải thiện các quy trình theo dõi dinh dưỡng hiệu quả hơn [3].

Vì vậy, bên cạnh việc lựa chọn thực phẩm sạch và lành mạnh, chúng ta cần có một phương pháp hiệu quả để tính toán lượng calo nạp vào cơ thể. Việc sử dụng công nghệ để hỗ trợ ước tính lượng calo từ hình ảnh món ăn có thể là một giải pháp hữu ích, giúp người tiêu dùng dễ dàng theo dõi chế độ dinh dưỡng, điều chỉnh khẩu phần ăn một cách chính xác và khoa học hơn.

Điều này không chỉ giúp duy trì một chế độ ăn uống cân bằng mà còn góp phần nâng cao sức khỏe lâu dài cho cộng đồng.

## 1.2. Cơ sở lý thuyết

### 1.2.1. Một số khái niệm cơ bản



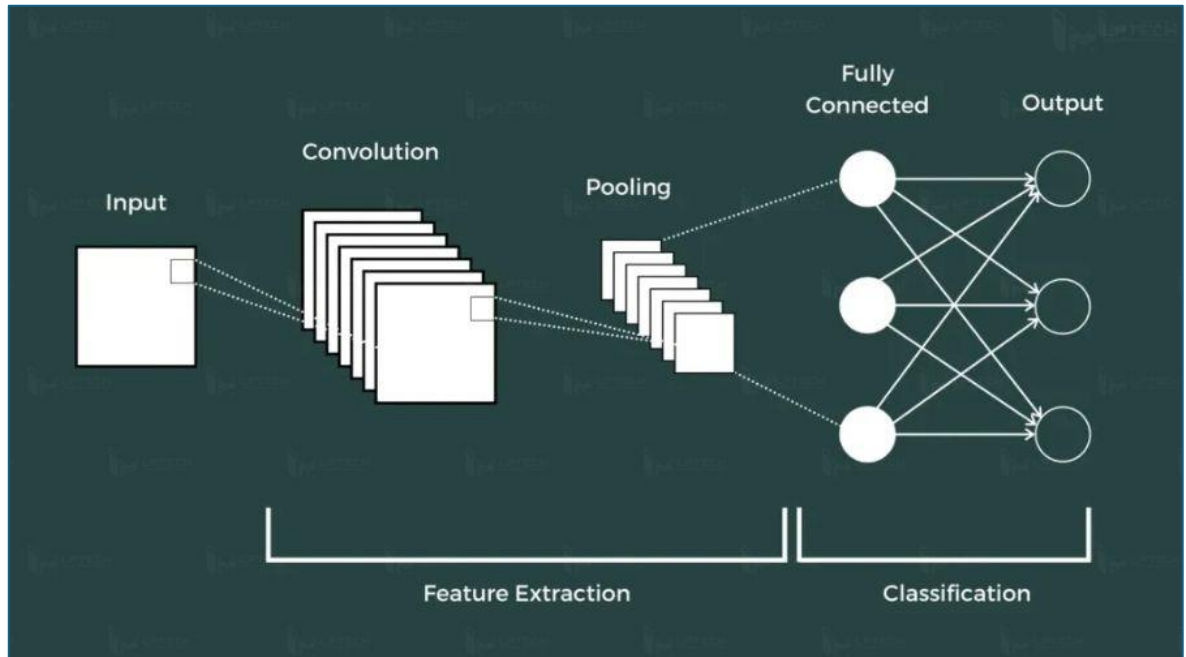
Hình 1.1 Mối quan hệ giữa trí tuệ nhân tạo, học máy và học sâu [4]

Trí tuệ nhân tạo (AI) là một cách tiếp cận làm cho máy móc suy nghĩ và hành xử một cách thông minh. Những cỗ máy này được điều khiển bởi phần mềm bên trong chúng, vì vậy, trí tuệ nhân tạo có liên quan nhiều đến các chương trình phần mềm thông minh điều khiển những máy móc này. AI là một ngành khoa học tìm kiếm các lý thuyết và phương pháp luận có thể giúp máy móc hiểu thế giới và phản ứng lại với các tình huống theo cách như con người vẫn làm [5].

Học máy (machine learning) là một lĩnh vực con của trí tuệ nhân tạo, trong đó máy tính có khả năng học từ dữ liệu mà không cần được lập trình rõ ràng. Thay vì viết mã cứng các quy tắc, thuật toán học máy sử dụng dữ liệu để tìm ra các mẫu và cải thiện hiệu suất theo thời gian.

Học sâu (deep learning) là một nhánh con của học máy, trong đó các mô hình sử dụng mạng nơ-ron nhân tạo nhiều lớp để học và trích xuất đặc trưng từ dữ liệu phức tạp.

### 1.2.2. Giới thiệu về mạng nơ-ron tích chập

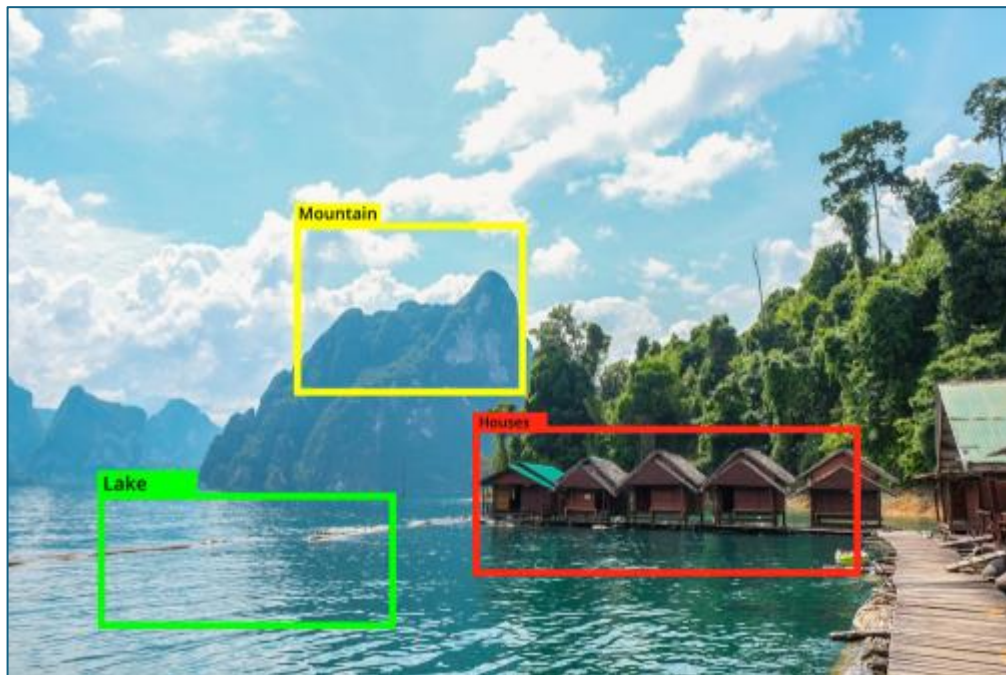


Hình 1.2 Kiến trúc CNN [6]

CNN là một dạng mô hình học sâu vô cùng mạnh mẽ trong lĩnh vực AI, nhất là trong việc xử lý hình ảnh. Nhờ vào việc nhận diện và phân tích những điểm quan trọng trong ảnh, nó sẽ ứng dụng vào những hệ thống thông minh một cách chính xác hơn [6].

CNN thường được ứng dụng trong việc giúp nhận diện khuôn mặt, phân loại đối tượng... CNN cũng đã được tích hợp trên Facebook, Google và nhiều nền tảng lớn khác nhằm giúp tăng khả năng nhận diện hình ảnh trên các dịch vụ của nền tảng [6].

### 1.2.3. Giới thiệu về phát hiện đối tượng



Hình 1.3 Hình ảnh ví dụ về nhận dạng đối tượng trong ảnh [7]

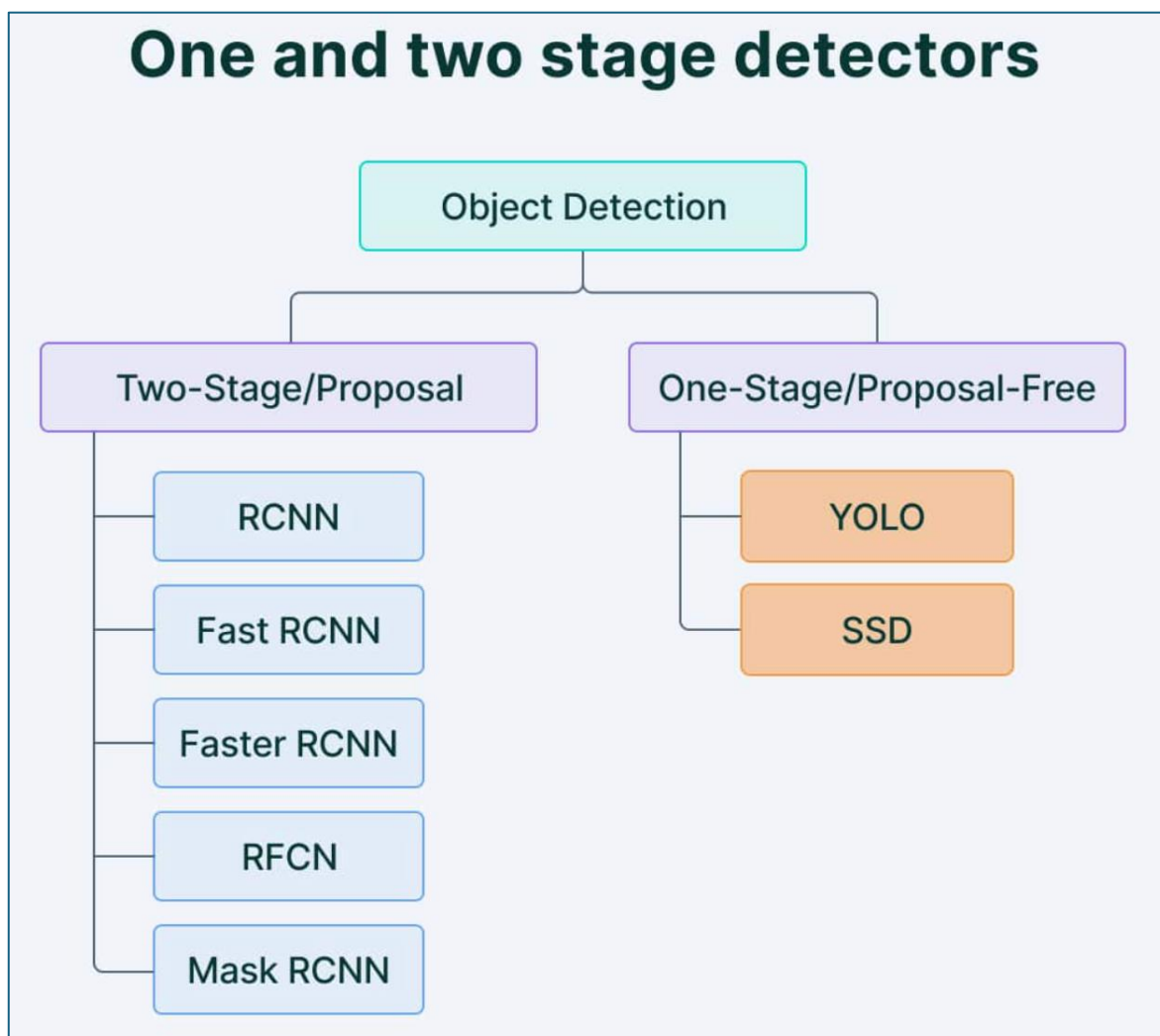
Phát hiện đối tượng là một nhiệm vụ lớn của lĩnh vực thị giác máy tính (computer vision), được coi là một trong những hướng phát triển quan trọng nhất của học sâu và xử lý hình ảnh (image processing).

Các mô hình phát hiện đối tượng thường được đào tạo để phát hiện sự hiện diện của các đối tượng cụ thể trong hình ảnh, video hoặc hoạt động thời gian thực (real time). Ngay cả trước khi có phương pháp học sâu và công nghệ xử lý hình ảnh hiện đại, việc phát hiện đối tượng vẫn được quan tâm rất nhiều. Khi đó người ta thường ứng dụng một số phương pháp xử lý ảnh như SIFT (Scale - Invariant Feature Transform) và HOG (Histogram of Oriented Gradients) với kỹ thuật trích xuất cạnh của vật thể [7].

Sự ra đời của CNN và những tiến bộ của công nghệ thị giác máy tính đã mang đến nhiều cách tiếp cận cũng như nhiều thuật toán phát hiện đối tượng hiệu quả hơn [7].

Các thuật toán phát hiện đối tượng có thể được chia thành hai loại chính: phát hiện đối tượng một giai đoạn (one-stage object detection) và phát hiện đối tượng hai giai đoạn (two-stage object detection).





Hình 1.4 Phân loại các thuật toán phát hiện đối tượng [8]

### **Phát hiện đối tượng một giai đoạn:**

Phát hiện đối tượng một giai đoạn sử dụng một lần truyền hình ảnh đầu vào để đưa ra dự đoán về sự hiện diện và vị trí của các đối tượng trong ảnh. Nó xử lý toàn bộ hình ảnh trong một lần chạy, khiến chúng trở nên hiệu quả về mặt tính toán [9].

Tuy nhiên, phát hiện đối tượng một giai đoạn thường kém chính xác hơn so với các phương pháp khác và kém hiệu quả hơn trong việc phát hiện các đối tượng nhỏ. Các thuật toán như vậy có thể được sử dụng để phát hiện các đối tượng theo thời gian thực trong các môi trường hạn chế về tài nguyên [9].

**Phát hiện đối tượng hai giai đoạn:** Phát hiện đối tượng hai giai đoạn sử dụng hai lần truyền hình ảnh đầu vào để đưa ra dự đoán về sự hiện diện và

vị trí của đối tượng. Lược đầu tiên được sử dụng để tạo một tập hợp các đề xuất hoặc vị trí tiềm năng của đối tượng. Lược thứ hai được sử dụng để tinh chỉnh các đề xuất này và đưa ra dự đoán cuối cùng. Cách tiếp cận này chính xác hơn Phát hiện đối tượng một giai đoạn nhưng cũng tốn kém hơn về mặt tính toán [9].

### **1.3. Phát biểu bài toán**

#### **1.3.1. Giới thiệu bài toán**

Bài toán ước lượng lượng calo trong bữa ăn từ hình ảnh là một bài toán thuộc lĩnh vực thị giác máy tính, cụ thể là bài toán phân loại với sự kết hợp giữa phát hiện đối tượng và phân đoạn hình ảnh. Với sự phát triển mạnh mẽ của trí tuệ nhân tạo và học sâu, việc xây dựng một hệ thống có khả năng nhận diện các món ăn từ hình ảnh và ước lượng lượng calo tương ứng đang trở thành một hướng nghiên cứu thực tiễn và mang lại nhiều lợi ích cho cộng đồng, đặc biệt là trong lĩnh vực chăm sóc sức khỏe và dinh dưỡng.

Hệ thống được thiết kế nhằm giúp người dùng theo dõi lượng calo tiêu thụ hàng ngày một cách thuận tiện, từ đó hỗ trợ họ xây dựng chế độ ăn uống hợp lý, phù hợp với mục tiêu cá nhân như giảm cân, duy trì vóc dáng, hoặc điều trị các bệnh liên quan đến chuyển hóa như tiểu đường, béo phì.

#### **1.3.2. Phát biểu bài toán**

Đầu vào của bài toán: Hình ảnh chụp món ăn thực tế với định dạng .jpg, sử dụng không gian màu RGB với 3 kênh màu. Trước khi được đưa vào hệ thống, ảnh sẽ trải qua một số bước tiền xử lý để phù hợp với định dạng và nâng cao độ chính xác của mô hình.

Đầu ra của bài toán: Thông tin đầu ra bao gồm:

- Danh sách các món ăn được phát hiện trong ảnh.
- Vị trí và hình dạng cụ thể của từng thành phần, được biểu diễn dưới dạng hộp giới hạn cho phép xác định vị trí của mỗi món ăn trong ảnh.
- Nhãn phân loại cho từng món ăn.

- Ước lượng lượng calo cho từng món ăn dựa trên thông tin nhận diện.
- Tổng lượng calo của toàn bộ bữa ăn sau khi tổng hợp thông tin từ các món riêng lẻ.

### 1.3.3. Các khó khăn và thách thức của bài toán

Bài toán ước lượng calo từ hình ảnh món ăn gặp phải nhiều khó khăn và thách thức lớn như:

- Đa dạng và phức tạp của bữa ăn: Một bữa ăn có thể bao gồm nhiều món ăn khác nhau, mỗi món lại được chế biến theo nhiều cách, với thành phần, hình dạng, màu sắc và cách bày trí rất đa dạng. Điều này gây khó khăn cho việc phát hiện, phân đoạn và phân loại chính xác từng thành phần trong ảnh.
- Ước lượng calo không chỉ dựa vào hình ảnh: Lượng calo của mỗi món ăn không chỉ phụ thuộc vào việc nhận diện loại thực phẩm, mà còn bị ảnh hưởng bởi khẩu phần ăn, khối lượng thực phẩm, phương pháp chế biến (chiên, hấp, luộc...), cũng như thành phần dinh dưỡng chi tiết. Những yếu tố này thường không thể hiện rõ ràng qua hình ảnh, làm cho việc ước lượng calo trở nên không chính xác nếu chỉ sử dụng thông tin thị giác.
- Thiếu dữ liệu chất lượng cao: Dữ liệu hình ảnh món ăn hiện nay còn hạn chế về số lượng và chất lượng, đặc biệt thiếu dữ liệu đã được gán nhãn đầy đủ (bao gồm hộp giới hạn, tên món ăn và lượng calo thực tế). Điều này ảnh hưởng trực tiếp đến hiệu quả huấn luyện của mô hình học sâu.
- Hạn chế về phần cứng và tài nguyên tính toán: Các mô hình học sâu hiện đại đòi hỏi phần cứng tính toán mạnh để đạt được hiệu năng cao trong cả giai đoạn huấn luyện và suy luận. Tuy nhiên, hệ thống được kỳ vọng có thể triển khai trên các thiết bị có cấu hình trung bình, nhằm tăng tính ứng dụng thực tiễn và khả năng tiếp cận cho người dùng phổ thông.

#### **1.3.4. Các miền ứng dụng của bài toán**

Bài toán ước lượng lượng calo từ ảnh món ăn có tiềm năng ứng dụng rộng rãi trong nhiều lĩnh vực, bao gồm:

- Y tế và chăm sóc sức khỏe: Hỗ trợ bác sĩ và bệnh nhân theo dõi chế độ ăn uống, đặc biệt với bệnh nhân mắc các bệnh như tiểu đường, béo phì, tim mạch.
- Ứng dụng cá nhân hóa dinh dưỡng: Kết hợp với các ứng dụng di động để cá nhân hóa kế hoạch dinh dưỡng, theo dõi calo hàng ngày, gợi ý món ăn phù hợp với mục tiêu cá nhân.
- Thể hình và thể thao: Giúp vận động viên và người tập thể thao kiểm soát lượng calo tiêu thụ để đạt mục tiêu tăng/giảm cân.
- Trường học và doanh trại: Kiểm soát và giám sát khẩu phần ăn cho học sinh, sinh viên và quân nhân một cách chính xác và hiệu quả.

## CHƯƠNG 2. MỘT SỐ KỸ THUẬT GIẢI QUYẾT BÀI TOÁN

### 2.1. Phương hướng tiếp cận bài toán

Bài toán ước lượng lượng calo trong bữa ăn từ hình ảnh là một thách thức liên ngành, kết hợp giữa thị giác máy tính, trí tuệ nhân tạo và khoa học dinh dưỡng. Mục tiêu chính là xây dựng một hệ thống thông minh có khả năng phân tích hình ảnh bữa ăn và đưa ra ước tính về tổng lượng calo chứa trong các món ăn được trình bày. Hệ thống này đóng vai trò hỗ trợ người dùng trong việc theo dõi và kiểm soát chế độ ăn uống, từ đó góp phần nâng cao sức khỏe cá nhân và hình thành thói quen ăn uống khoa học.

Phương hướng tiếp cận bài toán được triển khai thông qua một quy trình gồm nhiều bước liên tiếp, tận dụng các kỹ thuật hiện đại trong phân tích hình ảnh và xử lý dữ liệu. Cụ thể, quy trình bao gồm các bước chính như sau:

- Chuẩn hóa dữ liệu đầu vào: Dữ liệu hình ảnh ban đầu cần được xử lý để phù hợp với định dạng mà hệ thống có thể xử lý hiệu quả, bao gồm các thao tác như chuyển đổi định dạng, gán nhãn, và tổ chức cấu trúc dữ liệu.
- Nhận diện món ăn trong ảnh: Áp dụng các kỹ thuật phân tích ảnh để xác định các món ăn xuất hiện trong bữa ăn, kèm theo vị trí và khu vực tương ứng trong ảnh.
- Ước lượng khối lượng món ăn: Dựa vào đặc điểm hình ảnh như kích thước hoặc diện tích chiếm dụng, hệ thống sẽ thực hiện các phép tính nhằm ước lượng khối lượng thực phẩm, thường thông qua mô hình dự đoán được huấn luyện từ dữ liệu thực nghiệm.
- Tra cứu giá trị dinh dưỡng: Khi đã có tên món ăn và khối lượng tương ứng, hệ thống sẽ truy xuất thông tin dinh dưỡng từ cơ sở dữ liệu, cụ thể là năng lượng tính theo đơn vị calo ứng với mỗi đơn vị khối lượng.
- Tổng hợp kết quả và hiển thị: Toàn bộ dữ liệu thu thập được sẽ được tổng hợp để đưa ra ước lượng cuối cùng về tổng lượng calo trong bữa ăn, và trình bày kết quả một cách trực quan cho người dùng.

Cách tiếp cận tổng thể này không chỉ dừng lại ở việc phân tích nội dung hình ảnh, mà còn hướng đến việc xây dựng một giải pháp toàn diện, hỗ trợ cá nhân hóa chế độ ăn và quản lý dinh dưỡng hiệu quả. Đây là nền tảng quan trọng trong việc phát triển các ứng dụng chăm sóc sức khỏe dựa trên công nghệ.

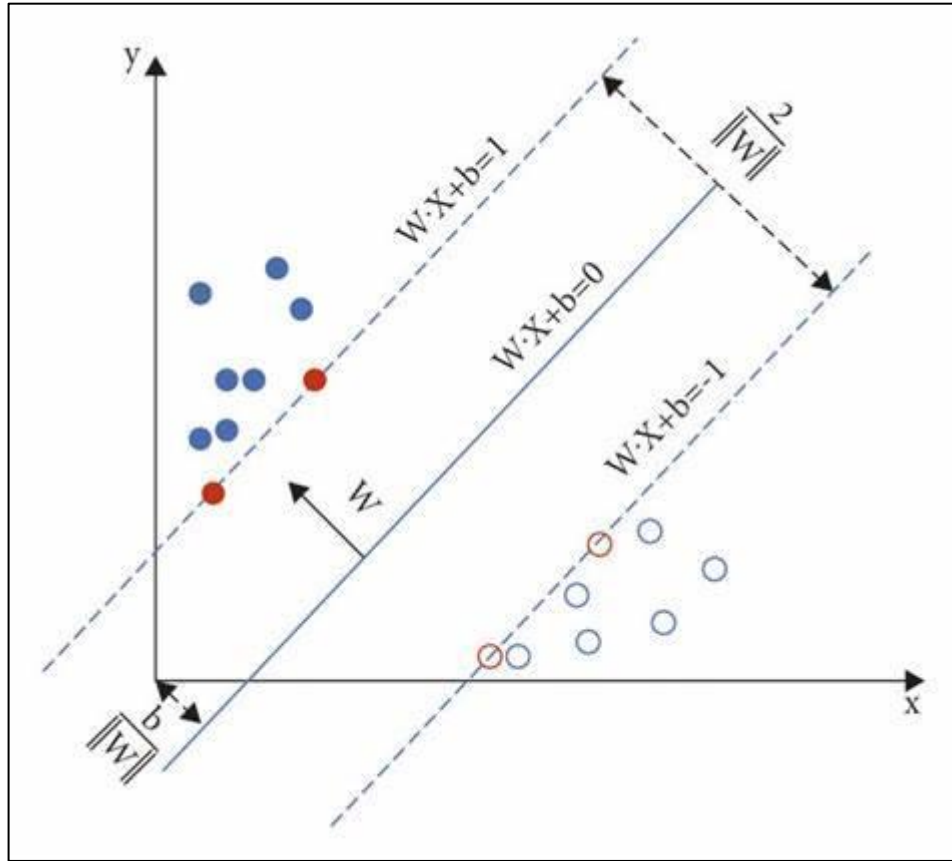
## **2.2. Một số kỹ thuật giải quyết bài toán**

Trong mục này, bài báo cáo sẽ giới thiệu một số kỹ thuật giải quyết bài toán ước lượng lượng calo trong bữa ăn từ hình ảnh. Các kỹ thuật này chủ yếu tập trung vào việc phát hiện và phân loại các món ăn trong bức ảnh. Quá trình này đóng vai trò quan trọng trong việc phân tách các đối tượng trong ảnh, đảm bảo sự nhận diện và phân loại chính xác cho từng món ăn.

### **2.2.1. Support Vector Machine**

SVM là một thuật toán học có giám sát, được sử dụng rộng rãi trong các bài toán phân lớp và hồi quy. Thuật toán SVM được phát triển bởi Vladimir Vapnik và các cộng sự vào thập niên 1990. Mục tiêu chính của SVM là tìm ra một siêu phẳng trong không gian đặc trưng có chiều cao tương ứng với số lượng đặc trưng của dữ liệu, sao cho siêu phẳng này phân chia dữ liệu thành hai phần, mỗi phần tương ứng với một lớp dữ liệu. Siêu phẳng này phải được tối ưu hóa sao cho khoảng cách giữa các điểm dữ liệu của hai lớp là lớn nhất, điều này giúp đảm bảo sự phân tách rõ ràng và chính xác giữa các lớp.

Từ góc độ đại số tuyến tính, siêu phẳng của SVM được xác định thỏa mãn: “lề cực đại”. “Lề cực đại” là khoảng cách giữa siêu phẳng và các điểm dữ liệu gần nhất (gọi là các vector hỗ trợ) của mỗi lớp được tối đa hóa. Điều này giúp tăng độ chính xác của mô hình, vì việc phân chia dữ liệu với lề lớn nhất giảm thiểu khả năng sai sót trong việc phân loại dữ liệu trong các tình huống chưa từng thấy.



Hình 2.1 Minh họa thuật toán SVM [10]

Sau đây là mô tả sơ lược về mô hình SVM. Xét bài toán phân lớp nhị phân, cho một tập dữ liệu huấn luyện gồm  $n$  mẫu, được biểu diễn dưới dạng:

$$X = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \in \mathbb{R}^{n \times d+1}$$

Trong đó,  $x_i$  là một véc tơ đặc trưng trong không gian  $\mathbb{R}^d$  và  $y_i \in \{-1, 1\}$  là nhãn lớp của mỗi mẫu dữ liệu. Mục tiêu của mô hình SVM là tìm ra một siêu phẳng tối ưu trong không gian đặc trưng để phân tách các lớp dữ liệu sao cho lề (margin) giữa hai lớp là lớn nhất. Siêu phẳng này có phương trình tổng quát là:

$$\langle w, x \rangle + b = 0$$

với  $w \in \mathbb{R}^d$  và  $b \in \mathbb{R}$ . Mục tiêu bài toán huấn luyện SVM là nhằm tìm ra siêu phẳng có lề cực đại, tức là tìm bộ tham số  $(w, b)$  sao cho khoảng cách giữa các mẫu dữ liệu và siêu phẳng phân tách là lớn nhất. Để thực hiện điều này, bài toán tối ưu sẽ được giải quyết dưới dạng:

$$\min_{w,b,\xi} \left\{ \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^n \xi_i \right\}$$

với các ràng buộc sau:

$$y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \begin{cases} \forall i = 1, \dots, n \\ \xi_i \geq 0, \forall i \end{cases}$$

Trong đó:

- $\langle ., . \rangle$  là phép nhân vô hướng trong không gian  $R^d$ .
- $\xi_i$  là các biến slack cho phép nới lỏng điều kiện phân lớp.
- $C$  là tham số điều chỉnh để điều tiết độ rộng của lề.

Một cách tiếp cận khác là giải bài toán đối ngẫu, giúp thay thế bài toán gốc bằng một dạng toán học tối ưu hóa có thể giải quyết hiệu quả hơn.

### **Ưu điểm:**

- Khả năng hoạt động hiệu quả trong không gian chiều cao: SVM cho phép xử lý dữ liệu có số chiều lớn, phù hợp với các bài toán như phân loại văn bản, nhận dạng hình ảnh,... trong đó mỗi mẫu có thể có hàng trăm hoặc hàng nghìn đặc trưng.
- Tối ưu hoá biên phân tách giữa các lớp: Mô hình SVM được huấn luyện để tìm ra siêu phẳng có khoảng cách lớn nhất đến các điểm dữ liệu gần nhất của mỗi lớp, giúp tăng khả năng khái quát hoá và giảm thiểu overfitting.
- Khả năng xử lý dữ liệu phi tuyến thông qua hàm kernel: Với kỹ thuật kernel, SVM có thể ánh xạ dữ liệu sang không gian đặc trưng có chiều cao hơn, từ đó giải quyết được các bài toán phân loại không tuyến tính.
- Hiệu quả trong trường hợp dữ liệu huấn luyện nhỏ: Do mô hình chỉ dựa vào một số lượng nhỏ các điểm hỗ trợ để xác định siêu phẳng phân tách nên SVM vẫn có thể hoạt động tốt với tập dữ liệu không quá lớn.

### **Nhược điểm:**

- Chi phí tính toán lớn khi xử lý tập dữ liệu có quy mô lớn: Việc giải bài toán tối ưu hoá trong SVM trở nên phức tạp và tốn tài nguyên khi số

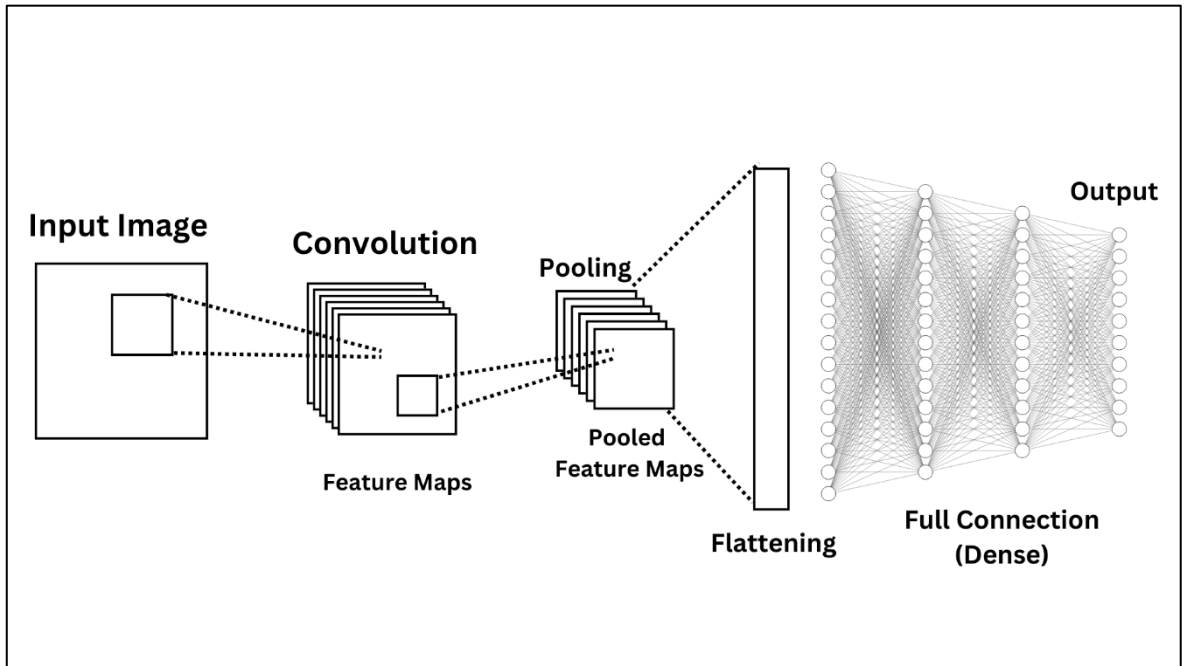


lượng mẫu huấn luyện tăng cao, đặc biệt khi sử dụng các kernel phi tuyến.

- Phụ thuộc vào việc lựa chọn kernel và tham số thích hợp: Hiệu quả của SVM phụ thuộc đáng kể vào việc lựa chọn hàm kernel và các siêu tham số như
- $C, \gamma, \dots$ . Việc lựa chọn không phù hợp có thể dẫn đến kết quả phân loại kém.
- Không cung cấp xác suất dự đoán trực tiếp: Mặc định, SVM chỉ đưa ra nhãn phân lớp mà không cung cấp độ tin cậy hoặc xác suất, điều này có thể gây hạn chế trong các bài toán cần ra quyết định dựa trên mức độ chắc chắn.
- Kém hiệu quả trong trường hợp dữ liệu nhiễu hoặc phân bố không rõ ràng: Trong các bài toán mà dữ liệu có nhiễu nhiễu hoặc không thể phân tách rõ ràng giữa các lớp, hiệu suất của SVM có thể bị suy giảm, ngay cả khi sử dụng cơ chế soft margin.

### 2.2.2. Mạng nơ-ron tích chập

Mạng nơ-ron tích chập (CNN) là một lớp mô hình học sâu được thiết kế đặc biệt để xử lý dữ liệu dạng lưới, điển hình là hình ảnh. Vào cuối những năm 1989, Yann LeCun đã phát triển mạng LeNet – một trong những CNN đầu tiên, để nhận diện chữ số viết tay trên tập dữ liệu MNIST [11]. Kể từ đó, CNN đã trở thành nền tảng cho nhiều ứng dụng trong lĩnh vực thị giác máy tính như phân loại ảnh, phát hiện đối tượng, nhận diện khuôn mặt và các hệ thống giám sát thông minh nhờ khả năng trích xuất đặc trưng cục bộ một cách tự động thông qua cơ chế học không cần thiết kế thủ công đặc trưng đầu vào.



Hình 2.2 Kiến trúc CNN [12]

CNN bao gồm 3 nhóm lớp chính, mỗi nhóm đảm nhiệm một chức năng cụ thể trong quy trình trích xuất và xử lý đặc trưng từ ảnh đầu vào:

- Lớp tích chập (convolutional layer): Đây là lớp đóng vai trò quan trọng nhất trong CNN, thực hiện chức năng trích xuất đặc trưng bằng cách sử dụng các bộ lọc (kernel) quét qua ảnh đầu vào. Mỗi bộ lọc học một mẫu đặc trưng như cạnh, góc hoặc họa tiết. Kết quả của phép tích chập là một bản đồ đặc trưng (feature map), phản ánh sự hiện diện của đặc trưng tại các vị trí khác nhau trong ảnh. Sau phép tích chập, các bản đồ thường được đưa qua hàm kích hoạt phi tuyến, chẳng hạn như ReLU, để tăng cường khả năng học các quan hệ phi tuyến phức tạp.
- Lớp gộp (pooling layer): Sau lớp tích chập, lớp gộp được sử dụng để giảm chiều không gian của các bản đồ đặc trưng, từ đó làm giảm số lượng tham số, giảm chi phí tính toán và cải thiện tính bất biến đối với các biến dạng nhỏ như dịch chuyển hoặc xoay nhẹ. Hai kỹ thuật phổ biến là max-pooling (lấy giá trị lớn nhất trong một vùng) và average-pooling (lấy giá trị trung bình).

- Lớp kết nối đầy đủ (fully connected layer): Sau quá trình trích xuất và gộp đặc trưng, các bản đồ đặc trưng được làm phẳng (flattening) thành một vector một chiều. Đây là bước trung gian giúp chuyển đổi dữ liệu từ dạng không gian sang dạng vector để có thể đưa vào các lớp kết nối đầy đủ. Tại đây, các đặc trưng được tổng hợp lại và học các quan hệ phức tạp hơn nhằm thực hiện tác vụ phân loại hoặc hồi quy. Trong bài toán phân loại nhiều lớp, lớp đầu ra thường sử dụng hàm softmax để đưa ra phân phối xác suất trên các lớp.

Trong quá trình huấn luyện CNN, lan truyền ngược (backpropagation) đóng vai trò quan trọng trong việc điều chỉnh trọng số của mô hình. Sau khi mạng thực hiện lan truyền thẳng và đưa ra dự đoán, sai số giữa đầu ra dự đoán và nhãn thực tế sẽ được tính toán thông qua một hàm mất mát. Đối với các bài toán phân loại, hàm mất mát phổ biến thường được sử dụng là cross-entropy, có công thức như sau:

$$L = - \sum_{i=0}^C y_i \log(\hat{y}_i)$$

Trong đó:

- $C$  là số lượng lớp.
- $y_i$  là giá trị thực tế.
- $\hat{y}_i$  là xác suất dự đoán của mô hình.

Sai số này sau đó được lan truyền ngược từ lớp đầu ra về các lớp trước đó thông qua việc tính đạo hàm của hàm mất mát theo từng trọng số trong mạng. Quá trình này cho phép mô hình học được cách điều chỉnh các trọng số sao cho giảm dần sai số dự đoán. Các thuật toán tối ưu như Stochastic Gradient Descent (SGD) hoặc Adam thường được sử dụng để cập nhật trọng số dựa trên gradient thu được. Quá trình huấn luyện được lặp lại qua nhiều vòng nhằm cải thiện độ chính xác của mô hình.

**Ưu điểm:**

- Khả năng trích xuất đặc trưng tự động: CNN có thể tự động học và trích xuất các đặc trưng quan trọng từ dữ liệu đầu vào, đặc biệt là hình ảnh, mà không cần can thiệp thủ công vào quá trình thiết kế đặc trưng như các phương pháp truyền thống.
- Hiệu quả cao trong xử lý dữ liệu hình ảnh: CNN đã chứng minh hiệu quả vượt trội trong nhiều bài toán thị giác máy tính như phân loại ảnh, phát hiện đối tượng, nhận diện khuôn mặt, và các ứng dụng y tế.
- Khả năng tổng quát hóa tốt: Nhờ kiến trúc nhiều lớp và khả năng học đặc trưng từ đơn giản đến phức tạp, CNN có khả năng tổng quát hóa cao đối với các dữ liệu chưa từng gặp trong quá trình huấn luyện.

#### **Nhược điểm:**

- Yêu cầu lượng lớn dữ liệu huấn luyện: CNN thường cần một tập dữ liệu huấn luyện lớn để đạt được hiệu suất tốt, đặc biệt là đối với các mạng sâu.
- Chi phí tính toán cao: Việc huấn luyện các mạng CNN lớn tiêu tốn nhiều tài nguyên tính toán, đòi hỏi phần cứng mạnh như GPU hoặc TPU.
- Khó giải thích: CNN hoạt động như một hệ thống hộp đen, rất khó để giải thích rõ ràng tại sao mô hình lại đưa ra một dự đoán cụ thể nào đó, điều này hạn chế khả năng ứng dụng trong các lĩnh vực đòi hỏi tính minh bạch cao.
- Dễ bị ảnh hưởng bởi dữ liệu nhiễu: Mặc dù CNN có khả năng chống lại một số biến dạng như dịch chuyển hoặc xoay nhẹ, nhưng vẫn có thể bị ảnh hưởng nếu dữ liệu chứa nhiễu hoặc các đặc điểm không điển hình.

### **2.2.3. YOLOv8**

#### **Tổng quan về YOLO:**

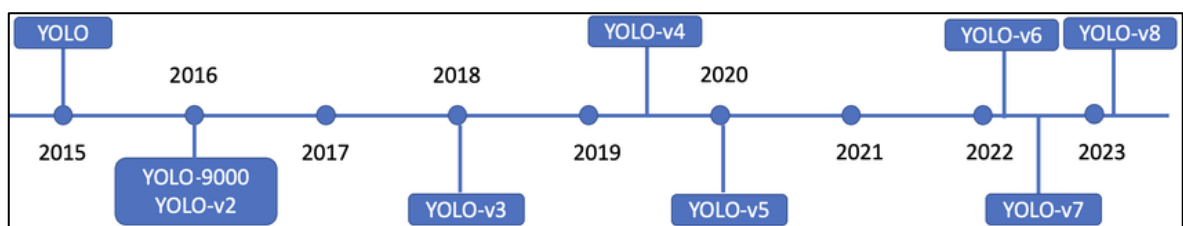
YOLO là một trong những kiến trúc mạng nơ-ron tiên phong, nổi tiếng trong lĩnh vực phát hiện đối tượng với độ chính xác cao và xử lý nhanh ở thời gian thực.

YOLO đề xuất sử dụng một mạng nơ-ron đầu-cuối (end-to-end neural network) để đồng thời đưa ra dự đoán về các hộp giới hạn và xác suất của các đối tượng. Phương pháp này khác biệt so với các cách tiếp cận truyền thống trong phát hiện đối tượng, vốn thường sử dụng các mô hình phân loại như một thành phần trung gian để thực hiện nhiệm vụ phát hiện.

Khác với các kỹ thuật trước đây, YOLO áp dụng một phương pháp tiếp cận cơ bản mới trong bài toán phát hiện đối tượng, qua đó đạt được những kết quả tiên tiến và vượt trội so với các thuật toán phát hiện đối tượng thời gian thực khác với một khoảng cách đáng kể.

Ngoài ra, các phương pháp sử dụng mạng đề xuất vùng yêu cầu thực hiện nhiều lần quét trên cùng một hình ảnh để phát hiện các đối tượng, trong khi đó YOLO hoàn thành toàn bộ quá trình dự đoán chỉ trong một lần duy nhất. Cách tiếp cận một lần (single shot/ one stage) này chính là yếu tố then chốt giúp YOLO đạt được tốc độ vượt trội trong các ứng dụng thời gian thực.

Kể từ lần ra mắt đầu tiên vào năm 2015, đã có nhiều phiên bản mới của mô hình YOLO được giới thiệu. Mỗi phiên bản đều được phát triển nhằm cải thiện những hạn chế của phiên bản trước đó, từ đó nâng cao hiệu suất và độ chính xác trong bài toán phát hiện đối tượng.



Hình 2.3 Mốc thời gian các phiên bản của YOLO tính tới năm 2023 [9]

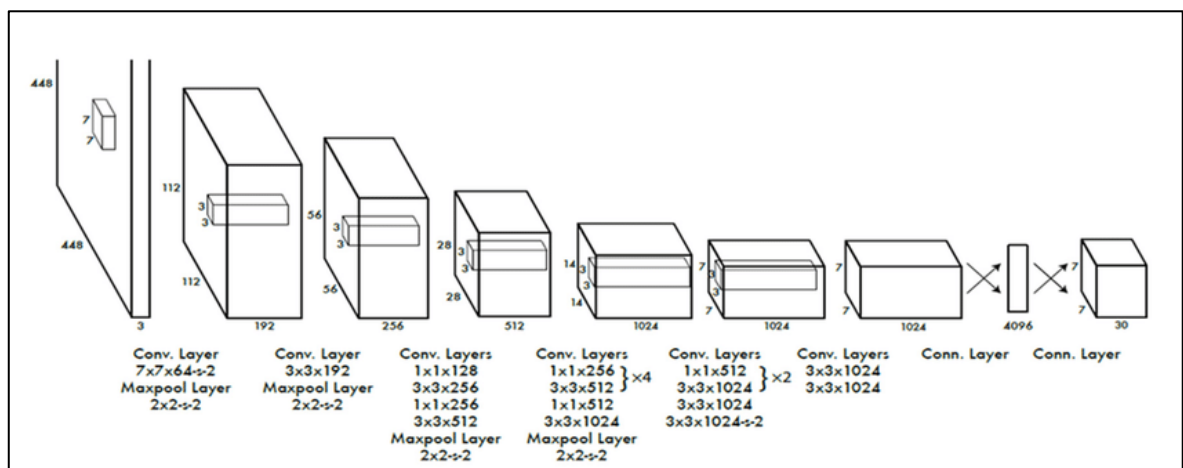
Hiện nay, bên cạnh những phiên bản đã được công bố như trên hình 2.3, YOLO đã phát triển thêm các phiên bản mới như YOLOv9, YOLOv10 và mới nhất là YOLOv11.

- YOLOv9 ứng dụng các phương pháp mới nhằm nâng cao hiệu quả huấn luyện và tối ưu hóa khả năng phát hiện đối tượng.

- YOLOv10 giới thiệu kiến trúc mới giúp loại bỏ nhu cầu sử dụng kỹ thuật Non-Maximum Suppression (NMS), cải thiện tốc độ và độ chính xác trong phát hiện thời gian thực.
- YOLOv11 là phiên bản mới nhất, đạt hiệu suất hàng đầu trên nhiều nhiệm vụ như phát hiện, phân đoạn, theo dõi và phân loại đối tượng, mở rộng khả năng ứng dụng trong các lĩnh vực trí tuệ nhân tạo.

### Kiến trúc và cách hoạt động:

Thuật toán YOLO lấy hình ảnh làm đầu vào, sau đó sử dụng mạng nơ-ron tích chập sâu đơn giản để phát hiện các đối tượng trong ảnh. Kiến trúc của mô hình CNN tạo thành xương sống của YOLO được hiển thị như hình bên dưới.



Hình 2.4 Kiến trúc CNN tạo thành xương sống của mô hình YOLO [9]

Trong giai đoạn tiền huấn luyện, 20 lớp tích chập đầu tiên của mô hình được huấn luyện trước trên tập dữ liệu ImageNet bằng cách kết nối tạm thời với một lớp tổng hợp trung bình (temporary average pooling layer) và một lớp kết nối đầy đủ. Sau khi hoàn tất quá trình tiền huấn luyện, mô hình này được điều chỉnh để phục vụ cho nhiệm vụ phát hiện đối tượng. Trong cấu trúc cuối cùng, lớp kết nối đầy đủ cuối cùng của YOLO chịu trách nhiệm dự đoán đồng thời cả xác suất của các lớp đối tượng và tọa độ của các hộp giới hạn.

YOLO chia hình ảnh đầu vào thành một lưới có kích thước  $S \times S$ . Nếu tâm của một đối tượng nằm trong một ô lưới bất kỳ, ô đó sẽ chịu trách nhiệm phát hiện đối tượng đó. Mỗi ô lưới sẽ dự đoán các hộp giới hạn B cùng với độ

tin cậy tương ứng cho mỗi hộp. Độ tin cậy phản ánh mức độ chắc chắn của mô hình rằng hộp giới hạn chứa một đối tượng, cũng như độ chính xác về việc hộp đó có khớp với đối tượng thực tế hay không.

Trong kiến trúc YOLO, mỗi ô lưới được thiết kế để có khả năng dự đoán nhiều hộp giới hạn. Tuy nhiên, trong quá trình huấn luyện, chỉ một hộp giới hạn dự đoán duy nhất được lựa chọn để đại diện cho mỗi đối tượng thực tế. Cụ thể, hộp được chọn là hộp giới hạn dự đoán có chỉ số giao nhau trên hộp (Intersection over Union – IoU) cao nhất so với hộp giới hạn thật (ground truth). Chiến lược này giúp phân công vai trò cụ thể cho từng bộ dự đoán, từ đó hình thành sự chuyên biệt hóa giữa các bộ hộp giới hạn. Nhờ vậy, mỗi bộ dự đoán trở nên tối ưu hơn trong việc nhận diện các đối tượng có đặc điểm hình học riêng biệt như kích thước, tỷ lệ khung hình hoặc kiểu dáng cụ thể. Cách tiếp cận này không chỉ giúp nâng cao khả năng học biểu diễn của mô hình mà còn cải thiện đáng kể recall tổng thể trong quá trình phát hiện đối tượng.

### **Các chỉ số đánh giá và kỹ thuật Non-Maximum Suppression:**

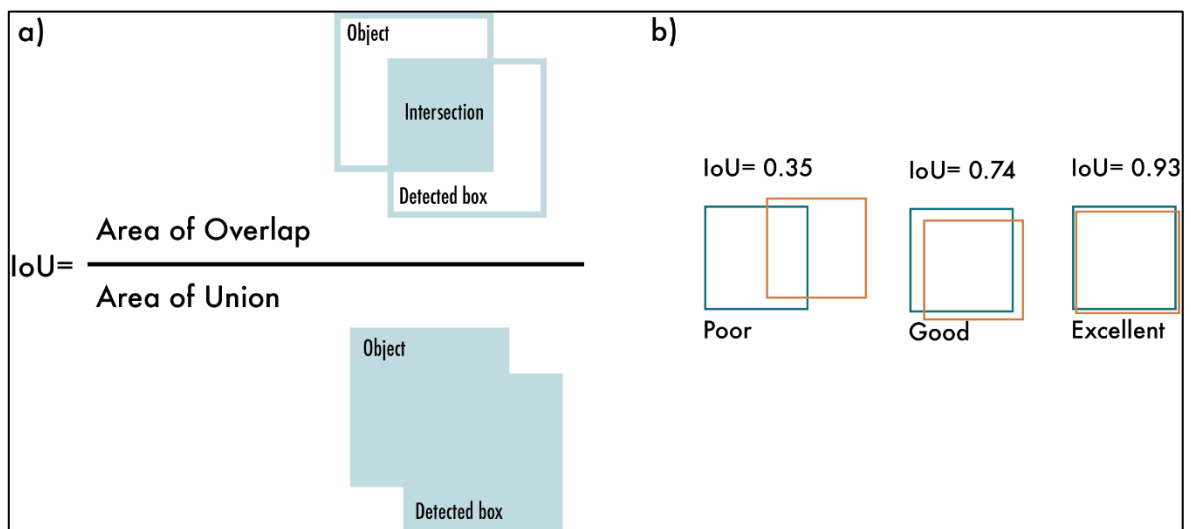
AP, mAP là chỉ số phổ biến nhất được sử dụng để đánh giá hiệu suất của các mô hình phát hiện đối tượng. Chỉ số này đo lường độ chính xác trung bình trên tất cả các danh mục đối tượng, cung cấp một giá trị duy nhất để so sánh giữa các mô hình khác nhau. Trong tập dữ liệu COCO, không có sự phân biệt giữa AP và mAP [13].

Chỉ số AP dựa trên các chỉ số precision-recall, xử lý nhiều danh mục đối tượng, và định nghĩa một dự đoán đúng dựa trên IoU:

- Precision và recall: precision đo lường độ chính xác của các dự đoán dương tính của mô hình, trong khi recall đo lường tỷ lệ các trường hợp thực sự dương tính mà mô hình dự đoán đúng. Giữa precision và recall thường tồn tại sự đánh đổi. Ví dụ, tăng số lượng đối tượng được phát hiện (tức tăng recall) có thể dẫn đến nhiều phát hiện sai (giảm precision). Để giải quyết sự đánh đổi này, chỉ số AP sử dụng đường

cong precision-recall, thể hiện mối quan hệ giữa precision và recall theo các ngưỡng độ tin cậy khác nhau. AP được tính bằng diện tích dưới đường cong precision-recall, cung cấp một đánh giá tổng thể cân bằng giữa precision và recall [13].

- Xử lý nhiều nhóm đối tượng: Các mô hình phát hiện đối tượng cần xác định và định vị nhiều loại đối tượng khác nhau trong một ảnh. Chỉ số AP giải quyết điều này bằng cách tính toán AP riêng cho từng nhóm, sau đó lấy trung bình các AP đó trên toàn bộ các nhóm (do đó gọi là mAP). Phương pháp này giúp đánh giá hiệu suất của mô hình một cách toàn diện và công bằng trên từng loại đối tượng [13].
- IoU: Bài toán phát hiện đối tượng nhằm mục tiêu định vị chính xác các đối tượng trong ảnh thông qua việc dự đoán các hộp giới hạn. Chỉ số AP sử dụng IoU để đánh giá chất lượng của các hộp dự đoán. IoU là tỷ lệ giữa diện tích giao nhau và diện tích hợp giữa hộp giới hạn dự đoán và hộp giới hạn thực tế. Nó đo lường mức độ trùng khớp giữa dự đoán và thực tế. Trong bộ đánh giá COCO, nhiều ngưỡng IoU khác nhau được sử dụng để đánh giá mô hình ở các mức độ chính xác định vị khác nhau [13].



Hình 2.5 Công thức và ví dụ về IoU [13]



NMS là một kỹ thuật xử lý hậu kỳ được sử dụng trong các thuật toán phát hiện đối tượng nhằm loại bỏ các hộp giới hạn chồng lặp không cần thiết và cải thiện chất lượng phát hiện tổng thể. Thông thường, các thuật toán phát hiện đối tượng tạo ra nhiều hộp giới hạn xung quanh cùng một đối tượng với các độ tin cậy khác nhau. NMS sẽ lọc bỏ các hộp dư thừa và không liên quan, chỉ giữ lại hộp có độ chính xác cao nhất. Thuật toán được mô tả như bên dưới:

```

0: Tập các hộp giới hạn được dự đoán B, độ tin cậy (confidence score) S,
   ngưỡng IoU  $\tau$ , ngưỡng tin cậy (confidence threshold) T
0: Tập các hộp được chọn sau khi lọc F
1:  $F \leftarrow \emptyset$ 
2: Filter the boxes:  $B \leftarrow \{b \in B | S(b) \geq T\}$ 
3: Sort the boxes B by their confidence scores in descending order
4: while  $B \neq \emptyset$  do
5:   Select the box b with the highest confidence score
6:   Add b to the set of final boxes F:  $F \leftarrow F \cup \{b\}$ 
7:   Remove b from the set of boxes B:  $B \leftarrow B - \{b\}$ 
8:   for all remaining boxes r in B do
9:     Calculate the IoU between b and r:  $iou \leftarrow IoU(b, r)$ 
10:    if  $iou \geq \tau$  then
11:      Remove r from the set of boxes B:  $B \leftarrow B - \{r\}$ 
12:    end if
13:  end for
14: end while

```



Hình 2.6 Kết quả sau khi áp dụng kỹ thuật NMS [13]

Mỗi hộp giới hạn dự đoán được mô tả bằng năm giá trị: tọa độ trung tâm của hộp giới hạn ( $x, y$ ), chiều rộng hộp ( $w$ ), chiều cao hộp ( $h$ ) và độ tin cậy. Độ tin cậy được tính bằng công thức sau:

$$\text{Confidence score} = Pr(\text{Objec}) * IoU(\text{pred}, \text{truth})$$

Trong đó,  $Pr(\text{Objec})$  là xác suất có một đối tượng xuất hiện trong hộp giới hạn;  $IoU(\text{pred}, \text{truth})$  là độ giao nhau giữa hộp giới hạn dự đoán và thực tế.

Mỗi ô lưới dự đoán một tập xác suất điều kiện đối tượng thuộc lớp nào trong danh mục các lớp là  $Pr(\text{Class}_i | \text{Object})$ . Độ tin cậy lớp (class confidence) xác định xác suất đối tượng thuộc về một lớp cụ thể, được tính như sau:

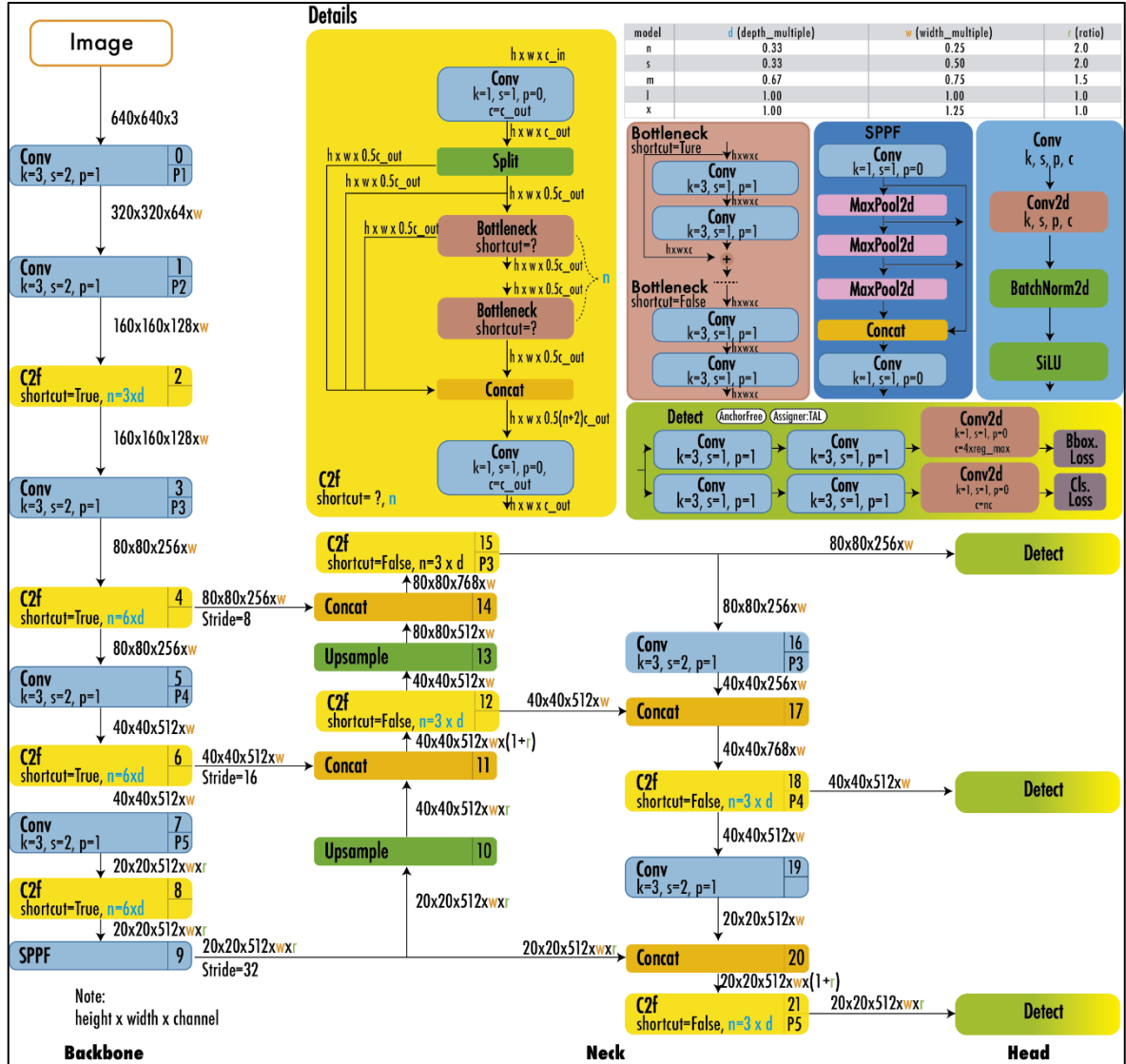
$$C_i = Pr(\text{Class}_i | \text{Object}) * Pr(\text{Objec}) * IoU(\text{pred}, \text{truth})$$

Hàm mất mát của YOLO là một hàm mất mát đa thành phần (multi-part loss function), kết hợp giữa: localization loss (tọa độ hộp giới hạn), confidence loss (mức độ tin cậy của hộp giới hạn), classification loss (xác suất lớp).

### Mô hình YOLOv8:

YOLOv8 được phát hành vào tháng 1 năm 2023 bởi Ultralytics, công ty đã phát triển YOLOv5. So với các phiên bản trước đó, YOLOv8 xuất hiện như một sự tiến hóa vượt bậc trong lĩnh vực phát hiện đối tượng. Với những cải tiến đáng kể về cấu trúc mạng, kỹ thuật huấn luyện, và khả năng dự đoán chính xác hơn, YOLOv8 đã giải quyết được nhiều hạn chế của các phiên bản

trước và cải thiện tốc độ cũng như độ chính xác. Những thay đổi này không chỉ giải quyết các hạn chế của những phiên bản trước mà còn mở ra các hướng đi mới, giúp mô hình đáp ứng tốt hơn trong các ứng dụng thực tiễn.



Hình 2.7 Kiến trúc YOLOv8 [13]

Hình 2.7 minh họa chi tiết kiến trúc của YOLOv8. YOLOv8 sử dụng backbone tương tự YOLOv5 với một số thay đổi ở CSPLayer, nay được gọi là module C2f. Module C2f (cross-stage partial bottleneck với hai lớp convolution) kết hợp các đặc trưng cấp cao với thông tin ngữ cảnh nhằm cải thiện độ chính xác phát hiện [13].

YOLOv8 sử dụng mô hình anchor-free với phần đầu ra (head) tách rời để xử lý độc lập các nhiệm vụ về xác suất có vật thể (objectness), phân loại và hồi quy (dự đoán tọa độ hộp giới hạn). Thiết kế này cho phép mỗi nhánh tập

trung vào nhiệm vụ riêng, từ đó cải thiện tổng thể độ chính xác của mô hình. Tại lớp đầu ra của YOLOv8, hàm sigmoid được sử dụng làm hàm kích hoạt cho objectness score, biểu diễn xác suất hộp giới hạn chứa một đối tượng. Hàm softmax được dùng để tính xác suất phân lớp, biểu diễn khả năng đối tượng thuộc từng lớp có thể có [13].

YOLOv8 sử dụng các hàm mất mát CIoU và DFL cho phần mất mát hộp giới hạn và hàm binary cross-entropy cho mất mát phân loại. Những hàm mất mát này đã cải thiện hiệu suất phát hiện vật thể, đặc biệt khi xử lý các vật thể nhỏ [13].

Thông tin và hiệu năng của các phiên bản (nano – n, nhỏ – s, vừa – m, lớn – l và rất lớn – x) trong họ YOLOv8 trên tập dữ liệu COCO được thể hiện qua bảng dưới đây.

*Bảng 2.1 Bảng thông tin hiệu suất của các phiên bản YOLOv8 trên tập dữ liệu COCO*

Model	size (pixels)	mAP <sup>val</sup> 50-95	params (M)	FLOPs (B)
YOLOv8n	640	37.3	3.2	8.7
YOLOv8s	640	44.9	11.2	28.6
YOLOv8m	640	50.2	25.9	78.9
YOLOv8l	640	52.9	43.7	165.2
YOLOv8x	640	53.9	68.2	257.8

Trong đó:

- Model: Tên phiên bản của mô hình.
- size (pixels): Kích thước ảnh đầu vào (640x640 pixels).
- mAP<sup>val</sup> 50-95: độ chính xác phát hiện đối tượng, với IoU từ 0.5 đến 0.95
- params (M): Số lượng tham số của mô hình (triệu).

- FLOPs (B): Số lượng phép tính dấu chấm động – đo độ phức tạp tính toán của mô hình (tỷ).

Ngoài nhiệm vụ phát hiện đối tượng, YOLOv8 còn hỗ trợ các nhiệm vụ khác trong lĩnh vực thị giác máy tính như: phân đoạn đối tượng, phát hiện tư thế, phân loại hình ảnh.

### 2.3. Hồi quy tuyến tính

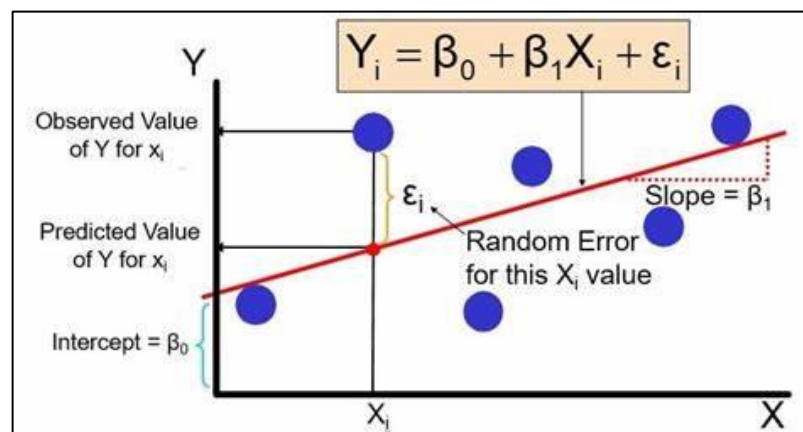
Bên cạnh vấn đề phát hiện món ăn trong bữa ăn, để có thể ước lượng được khối lượng của từng món cần sử dụng một số kỹ thuật khác để phụ trợ, trong đó đặc biệt là kỹ thuật hồi quy tuyến tính.

#### Giới thiệu:

Hồi quy tuyến tính (linear regression) là một phương pháp thống kê được sử dụng để phân tích và dự đoán dữ liệu, trong đó biến phụ thuộc có giá trị liên tục, còn các biến độc lập có thể là giá trị liên tục hoặc giá trị phân loại. Nói cách khác, hồi quy tuyến tính là công cụ dự đoán giá trị của biến phụ thuộc (Y) dựa trên giá trị của biến độc lập (X).

Phương pháp này đặc biệt hữu ích trong các tình huống cần dự đoán một lượng liên tục. Một số ứng dụng điển hình của hồi quy tuyến tính bao gồm: dự đoán lưu lượng khách hàng tại một cửa hàng bán lẻ, ước tính thời gian người dùng dành trên một trang web cụ thể, ước lượng giá nhà dựa vào diện tích,...

#### Cơ chế hoạt động:



Hình 2.8 Minh họa thuật toán hồi quy tuyến tính [14]

Hồi quy tuyến tính sử dụng một phương trình tuyến tính để biểu diễn mối quan hệ giữa biến phụ thuộc (Y) và các biến độc lập (X). Phương trình có dạng:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \epsilon$$

Trong đó:

- Y: biến phụ thuộc.
- $X_i$ : các biến độc lập.
- $\beta_0$ : hệ số chặn.
- $\beta_i$ : các hệ số hồi quy.
- $\epsilon$ : thành phần nhiễu.

Các hệ số hồi quy  $\beta_0, \beta_1, \beta_n$  được ước lượng bằng phương pháp bình phương tối thiểu (Ordinary Least Squares - OLS). Mục tiêu của phương pháp này là tìm các giá trị của các tham số sao cho tổng bình phương sai số giữa giá trị thực tế và giá trị dự đoán là nhỏ nhất.

Sau khi xây dựng mô hình, chất lượng của nó được đánh giá thông qua các chỉ số như:

- $R^2$  (hệ số xác định): Đánh giá mức độ giải thích của mô hình đối với biến phụ thuộc.
- Sai số chuẩn (standard error): Xác định mức độ chênh lệch trung bình giữa giá trị dự đoán và giá trị thực tế.

Khi mô hình đã được xây dựng và đánh giá, nó có thể được sử dụng để dự đoán giá trị của biến phụ thuộc dựa trên các giá trị mới của biến độc lập.

#### **Ưu điểm:**

- Đơn giản và dễ hiểu: Hồi quy tuyến tính có cách tiếp cận đơn giản và trực quan, dễ dàng giải thích mối quan hệ giữa biến phụ thuộc và các biến độc lập thông qua phương trình tuyến tính.
- Tính toán nhanh: Với các dữ liệu vừa và nhỏ, hồi quy tuyến tính hoạt động nhanh chóng nhờ các thuật toán tối ưu hóa như bình phương tối thiểu (OLS).

- Phân tích mối quan hệ: Phương pháp này cung cấp cái nhìn rõ ràng về mức độ ảnh hưởng của từng biến độc lập đến biến phụ thuộc thông qua các hệ số hồi quy.
- Khả năng mở rộng: Có thể dễ dàng mở rộng từ hồi quy tuyến tính đơn (một biến độc lập) sang hồi quy tuyến tính đa biến (nhiều biến độc lập).
- Công cụ phân tích phổ biến: Được hỗ trợ bởi nhiều thư viện trong các ngôn ngữ lập trình phổ biến như Python (Statsmodels, Scikit-learn), R, MATLAB,... giúp dễ dàng triển khai và áp dụng.

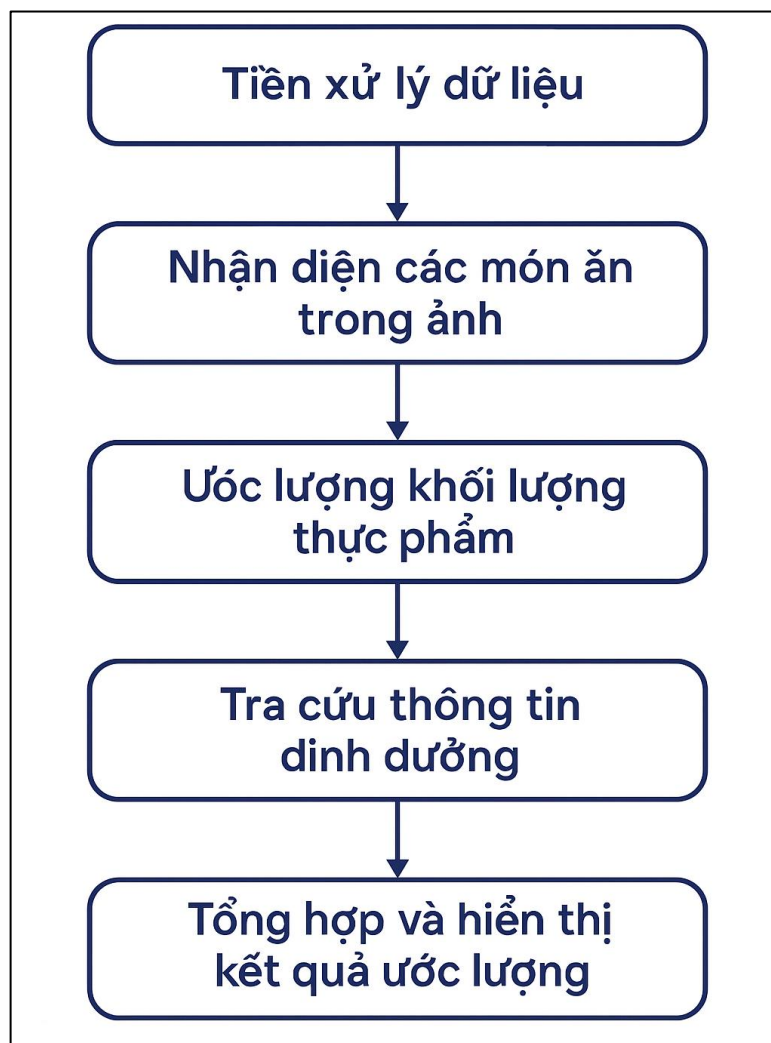
**Nhược điểm:**

- Giả định tuyến tính: Hồi quy tuyến tính giả định rằng mối quan hệ giữa biến phụ thuộc và các biến độc lập là tuyến tính, điều này có thể không phản ánh đúng thực tế khi mối quan hệ phức tạp hơn.
- Nhạy cảm với dữ liệu ngoại lai: Các giá trị ngoại lai có thể làm sai lệch kết quả, ảnh hưởng đáng kể đến độ chính xác của mô hình.
- Đòi hỏi độc lập giữa các biến: Mô hình không hoạt động hiệu quả nếu các biến độc lập có mối quan hệ đa cộng tuyến, vì điều này làm giảm khả năng ước lượng chính xác các hệ số hồi quy.
- Không phù hợp với dữ liệu phi tuyến: Hồi quy tuyến tính không hoạt động tốt với dữ liệu có mối quan hệ phi tuyến giữa các biến.
- Phụ thuộc vào phân phối sai số: Mô hình giả định rằng các sai số tuân theo phân phối chuẩn với phương sai không đổi. Nếu giả định này không được đáp ứng, hiệu suất của mô hình sẽ bị giảm sút.
- Giới hạn trong dự đoán phi tuyến: Hồi quy tuyến tính không thể giải quyết hiệu quả các bài toán phức tạp yêu cầu dự đoán phi tuyến hoặc xử lý dữ liệu có tương tác phức tạp giữa các biến.

## CHƯƠNG 3. THỰC NGHIỆM

### 3.1. Quy trình thực nghiệm

Nhằm triển khai giải pháp ước lượng lượng calo trong bữa ăn từ hình ảnh một cách hiệu quả, quy trình thực nghiệm trong bài báo cáo bao gồm các bước được tổ chức tuần tự và logic. Mỗi bước trong quy trình được thiết kế để hỗ trợ tối đa cho bước kế tiếp, đảm bảo tính nhất quán và khả năng mở rộng của hệ thống. Cụ thể, quy trình thực nghiệm được chia thành năm giai đoạn chính như sau:



Hình 3.1 Quy trình thực nghiệm

- **Bước 1:** Tiền xử lý dữ liệu: Dữ liệu đầu vào bao gồm hình ảnh và các thông tin liên quan được chuẩn hóa nhằm phù hợp với yêu cầu của các mô hình học sâu. Quá trình này bao gồm gán nhãn dữ liệu, chuyển đổi



định dạng, và tổ chức lại cấu trúc thư mục. Đây là bước khởi đầu mang tính quyết định đến chất lượng huấn luyện mô hình.

- **Bước 2:** Nhận diện các thành phần món ăn trong ảnh: Sau khi dữ liệu được chuẩn hóa, hệ thống tiến hành nhận diện các món ăn có trong hình ảnh thông qua mô hình học sâu. Kết quả đầu ra của bước này là các đối tượng món ăn được xác định vị trí và phân loại, làm cơ sở cho việc phân tích chi tiết ở các bước tiếp theo.
- **Bước 3:** Ước lượng khối lượng thực phẩm: Dựa trên vùng phát hiện của từng món ăn, hệ thống trích xuất các đặc trưng hình ảnh và tiến hành dự đoán khối lượng món ăn thông qua mô hình hồi quy. Mô hình được huấn luyện từ dữ liệu mẫu với mối liên hệ giữa thể tích và khối lượng, giúp đưa ra ước lượng chính xác và hợp lý.
- **Bước 4:** Tra cứu thông tin dinh dưỡng: Sau khi có được tên món ăn và khối lượng tương ứng, hệ thống tra cứu dữ liệu dinh dưỡng để thu thập thông tin năng lượng (tính theo kcal trên mỗi 100g) của từng món ăn. Từ đó, lượng calo của mỗi món được tính toán dựa trên khối lượng đã dự đoán.
- **Bước 5:** Tổng hợp và hiển thị kết quả ước lượng: Toàn bộ thông tin thu được từ các bước trên được tổng hợp để tính toán lượng calo tổng thể trong bữa ăn. Kết quả được trình bày dưới dạng trực quan, hỗ trợ người dùng trong việc theo dõi và điều chỉnh chế độ ăn uống của mình một cách hiệu quả.

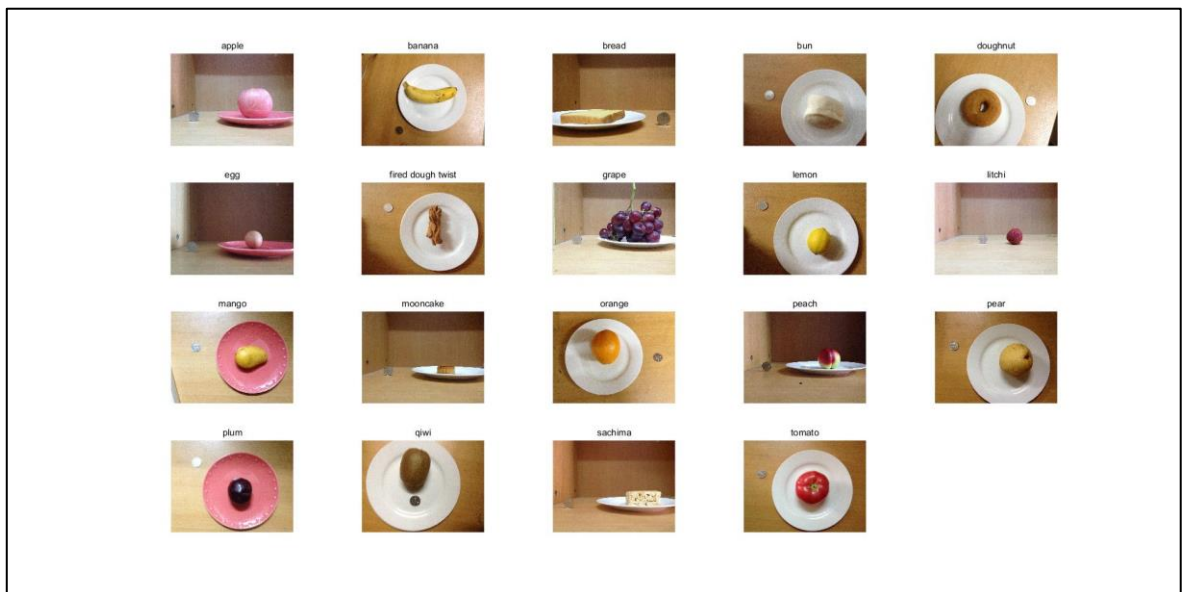
Quy trình thực nghiệm trên không chỉ đảm bảo tính khả thi trong triển khai thực tế, mà còn tạo nền tảng vững chắc cho các bước trình bày chi tiết tiếp theo trong chương này, bao gồm: tiền xử lý dữ liệu, huấn luyện mô hình, và đánh giá kết quả huấn luyện.

### 3.2. Dữ liệu thực nghiệm

Hiện nay, các tập dữ liệu hình ảnh thực phẩm hiện có chỉ tập trung vào việc phân loại loại thực phẩm, mà không cung cấp thông tin định lượng như

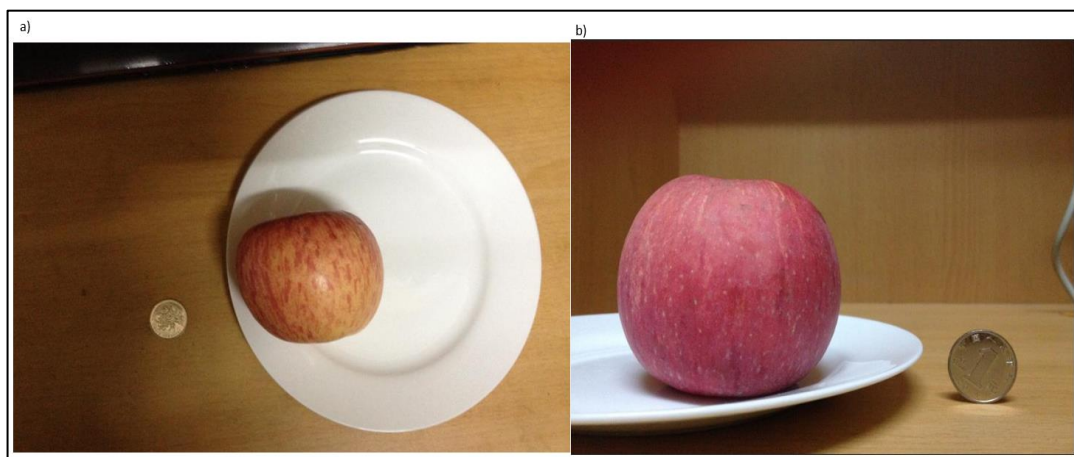
khối lượng hay thể tích — điều này gây khó khăn cho các bài toán ước lượng calo. Để khắc phục hạn chế này, nhóm nghiên cứu tại Đại học Khoa học và Công nghệ Đông Trung Hoa (ECUST) đã xây dựng bộ dữ liệu ECUST Food Dataset (ECUSTFD) với mục tiêu hỗ trợ trực tiếp cho các ứng dụng ước lượng dinh dưỡng bằng học sâu.

ECUSTFD là một tập dữ liệu công khai miễn phí, bao gồm 2.978 ảnh của 19 loại thực phẩm khác nhau như táo, trứng, cà chua, chuối, bánh mì,... Mỗi ảnh trong tập dữ liệu đều được chụp trong điều kiện ánh sáng tự nhiên, có kèm theo một vật tham chiếu kích thước để làm cơ sở tính toán thể tích món ăn. Tập dữ liệu đi kèm với file nhãn vị trí hộp giới hạn của thực phẩm trong ảnh (định dạng .xml theo chuẩn Pascal VOC), cũng như các thông tin về khối lượng, thể tích thực tế và giá trị calo của từng đối tượng.



*Hình 3.2 Một số hình ảnh trong bộ dữ liệu*

Đối với mỗi một phần thức ăn duy nhất, bức ảnh được chụp bằng điện thoại thông minh, với hai góc nhìn chính là góc nhìn từ trên xuống và góc nhìn từ bên cạnh. Tập dữ liệu ECUSTFD đã được tiền xử lý và chia tỷ lệ lại trong phiên bản mở rộng ECUSTFD-resized [15], giúp thuận tiện hơn cho việc huấn luyện các mô hình học sâu hiện đại. Các hình ảnh trong tập dữ liệu có độ phân giải đa dạng, tuy nhiên tất cả đều có kích thước dưới  $1000 \times 1000$  pixels, đảm bảo phù hợp cho việc xử lý và huấn luyện mô hình học sâu.



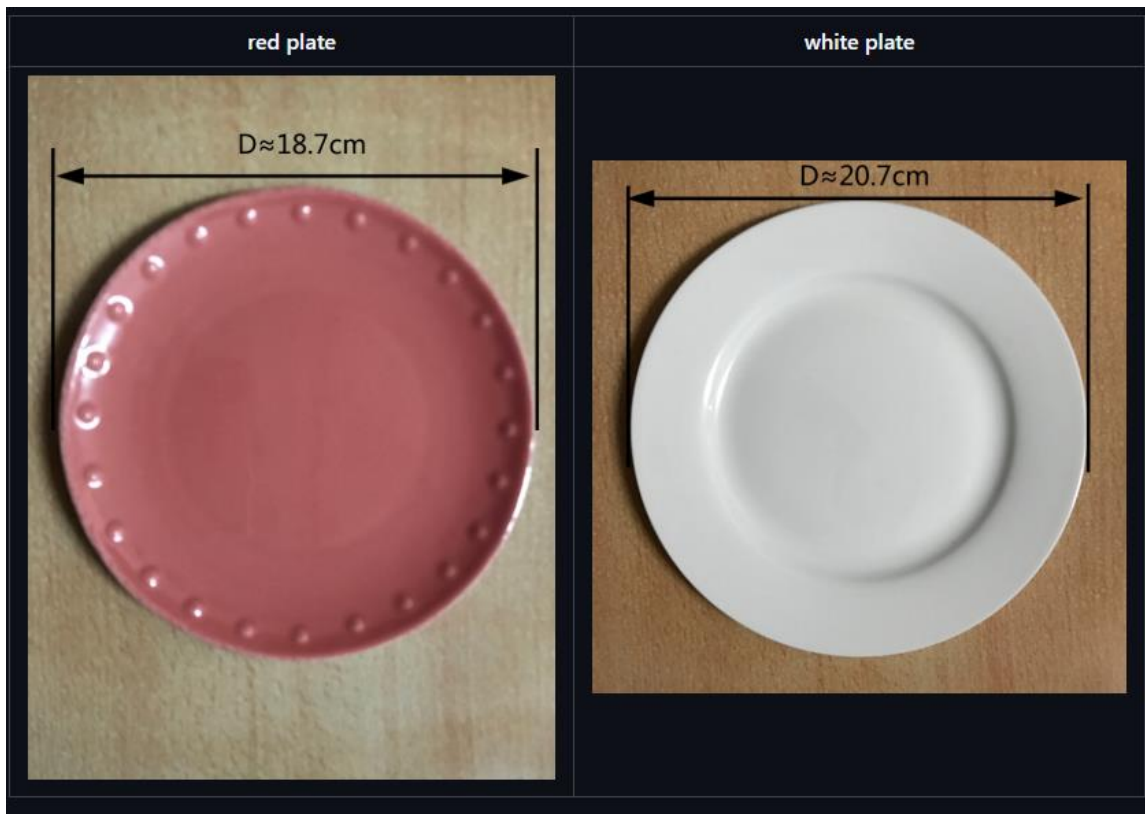
*Hình 3.3 Minh họa hai góc nhìn trong bộ dữ liệu*

Đồng xu là một vật tham chiếu luôn xuất hiện trong mỗi bức ảnh của bộ dữ liệu. Do đó, đồng xu sẽ được gắn nhãn và làm vật tham chiếu để thực hiện tính toán kích thước cho đối tượng trong ảnh. Đồng xu có kích thước đường kính 25.0 mm.



*Hình 3.4 Kích thước vật tham chiếu đồng xu*

Ngoài ra, một số bức ảnh được chụp với đĩa trắng và đĩa đỏ có tác dụng làm vật tham chiếu tương đương đồng xu. Đường kính của đĩa đỏ là 18.7 cm, chiều cao là 2.0 cm. Còn với đĩa trắng, đường kính là 20.7 cm và chiều cao cũng là 2.0 cm.



Hình 3.5 Kích thước vật tham chiếu đĩa đỏ và đĩa trắng

Bộ dữ liệu ECUSTFD-resized còn cung cấp một tệp excel chứa các thông tin về thể tích và khối lượng mỗi món ăn.

	A	B	C	D	E	F	G	H	I	J
1	id	type	volume (cm <sup>3</sup> )	weight (g)						
2	apple001	apple	310	244, 5						
3	apple002	apple	290	232, 5						
4	apple003	apple	280	219						
5	apple004	apple	300	234						
6	apple005	apple	280	212, 5						
7	apple006	apple	290	238						
8	apple007	apple	420	325						
9	apple008	apple	300	238						
10	apple009	apple	360	288, 5						
11	apple010	apple	330	260, 5						
12	apple011	apple	310	230						
13	apple012	apple	310	246						
14	apple013	apple	350	252, 5						
15	apple014	apple	290	236						
16	apple015	apple	390	293, 5						
17	apple016	apple	330	272, 5						
18	apple017	apple	330	255						
19	apple018	apple	290	231, 5						
20	apple019	apple	300	255						
21										
22										
23										
24										
25										
26										

apple banana bread bun doughnut egg fired\_dough\_twist grape lemon litchi mango mooncake orange ... + :

Hình 3.6 Hình minh họa dữ liệu có trong trang tính “apple”

### 3.3. Tiền xử lý dữ liệu

Bộ dữ liệu ECUSTFD-resized lưu thông tin các nhãn dưới định dạng tệp XML(Extensible Markup Language) theo định dạng PASCAL VOC. Tuy nhiên để huấn luyện mô hình YOLOv8, dữ liệu cần được chuyển đổi sang định dạng YOLO, trong đó thông tin hộp giới hạn, lớp của các đối tượng trong một ảnh được lưu trong một tệp văn bản .txt theo cấu trúc: class\_id x\_center y\_center width height (các thông tin được chuẩn hóa theo kích thước hình ảnh).

Do đó, bài báo cáo đã được xây dựng một quy trình tiền xử lý dữ liệu, bao gồm các bước chính: đọc và trích xuất thông tin từ các tệp XML (tên lớp, tọa độ hộp giới hạn, kích thước ảnh), chuyển đổi tọa độ từ định dạng VOC sang định dạng YOLO, ánh xạ tên lớp sang chỉ số lớp, và lưu các tệp nhãn dưới dạng .txt tương ứng với từng ảnh đầu vào.

Tất cả các đoạn mã thực thi việc xử lý dữ liệu được thực hiện trên môi trường phát triển tích hợp Visual Studio Code, Pycharm với python phiên bản 3.10.16. Một số thông tin cơ bản của máy gồm có:

- Hệ điều hành: Windows 11 Home Single Language (64-bit).
- CPU (Central Processing Unit): 11th Gen Intel Core i5-11300H.
- RAM (Random Access Memory): 16GB.
- GPU (Graphics Processing Unit): NVIDIA GeForce GTX 1650.

Đầu tiên, thực hiện chia tập dữ liệu gồm 2978 ảnh kèm các tệp XML thành 3 tập: training set, validation set và test set với tỉ lệ tương ứng là 70%, 20% và 10%.

```
D:\conda\envs\myenv\python.exe D:\TaiLieuHocTap\BATN\food-calo-project\scripts-process-data\split_dataset.py
Copying train: 100%|██████████| 2084/2084 [00:27<00:00, 74.81it/s]
Copying val: 100%|██████████| 596/596 [00:07<00:00, 83.25it/s]
Copying test: 100%|██████████| 298/298 [00:04<00:00, 74.40it/s]
✅ Done splitting: 2084 train, 596 val, 298 test.
```

*Hình 3.7 Số lượng ảnh trong mỗi tập*

Tiếp theo, tiến hành chuyển đổi các tệp định dạng XML sang các tệp văn bản .txt.

The image shows a code editor with two tabs: 'apple001S(1).xml' and 'apple001S(1).txt'. The XML tab contains the following code:

```

19 <segmented>0</segmented>
20 <object>
21   <name>apple</name>
22   <pose>Unspecified</pose>
23   <truncated>0</truncated>
24   <difficult>0</difficult>
25   <bndbox>
26     <xmin>326</xmin>
27     <ymin>229</ymin>
28     <xmax>540</xmax>
29     <ymax>401</ymax>
30   </bndbox>
31 </object>
32 <object>
33   <name>coin</name>
34   <pose>Unspecified</pose>
35   <truncated>0</truncated>
36   <difficult>0</difficult>
37   <bndbox>
38     <xmin>196</xmin>
39     <ymin>344</ymin>
40     <xmax>255</xmax>
41     <ymax>395</ymax>

```

The 'apple001S(1).txt' tab shows a table with 3 rows and 6 columns:

1	1	0.530637	0.571688	0.262255	0.312160
2	0	0.276348	0.670599	0.072304	0.092559
3					

Hình 3.8 Kết quả tệp văn bản sau khi chuyển đổi tệp XML tương ứng

Cấu hình tệp “data.yaml” mô tả cấu trúc dữ liệu đầu vào cho mô hình YOLOv8 gồm đường dẫn tới tập huấn luyện, tập xác thực và tập kiểm tra; số lớp và tên các lớp.

The image shows a code editor with a tab labeled 'data.yaml'. The file contains the following configuration:

```

1 train: /content/drive/MyDrive/data/final-project/food-calor-project/dataset/images/train
2 val: /content/drive/MyDrive/data/final-project/food-calor-project/dataset/images/val
3 test: /content/drive/MyDrive/data/final-project/food-calor-project/dataset/images/test
4
5 nc: 20
6 names:
7   - coin
8   - apple
9   - banana
10  - bread
11  - bun
12  - doughnut
13  - egg
14  - fired_dough_twist
15  - grape
16  - lemon
17  - litchi
18  - mango
19  - mooncake
20  - orange
21  - pear
22  - peach
23  - plum
24  - qiwi
25  - sachima
26  - tomato

```

Hình 3.9 Thông tin cấu hình tệp data.yaml

### 3.4. Huấn luyện mô hình

#### 3.4.1. YOLOv8

Việc huấn luyện mô hình học sâu YOLOv8 được thực hiện trên Google Colab với ngôn ngữ lập trình python cùng GPU NVIDIA Tesla T4 - được google colab cấp miễn phí, nhằm tăng tốc xử lý, rút ngắn thời gian huấn luyện và đảm bảo hiệu suất tính toán.

Trong quá trình huấn luyện, phiên bản nano của mô hình YOLOv8 (yolov8n.pt) được sử dụng để phù hợp với thời gian sử dụng GPU trên Google Colab. Lệnh huấn luyện được sử dụng như sau:

```
[ ] # Path
project_path = "/content/drive/MyDrive/data/final-project/food-calo-project"
dataset_path = f"{project_path}/dataset"
save_dir = f"{project_path}/runs"

# Load model
model = YOLO('yolov8n.pt')

# Train
model.train(
    data=f'{dataset_path}/data.yaml',
    epochs=50,
    imgsz=640,
    batch=32,
    project=save_dir,
    name='food_calo_project',
    device=0,
    patience=10,
    save_period=10,
    workers=2,
    optimizer='AdamW',
    cos_lr=True,
    augment=True
)
```

Hình 3.10 Lệnh huấn luyện mô hình yolov8n

Trong đó:

- data=f'{dataset\_path}/data.yaml': xác định đường dẫn đến tệp cấu hình bộ dữ liệu data.yaml.
- epochs=50: số lượng vòng lặp huấn luyện.



- `imgsz=640`: thay đổi kích thước ảnh đầu vào sao cho phù hợp với kích thước đầu vào của YOLO (640x640).
- `batch=32`: số lượng ảnh đưa vào mô hình mỗi lần huấn luyện.
- `project=save_dir`: xác định đường dẫn đến thư mục lưu kết quả huấn luyện.
- `name='food_calor_project'`: xác định tên thư mục lưu trữ trọng số của mô hình.
- `device=0`: chỉ định gpu là thiết bị huấn luyện.
- `patience=10`: số epoch tối đa không cải thiện trước khi dừng sớm khi `loss/accuracy` không cải thiện.
- `save_period=10`: lưu trọng số mô hình sau mỗi 10 epoch.
- `workers=2`: xác định số luồng xử lý để tăng tốc độ load dữ liệu trong khi huấn luyện (Google Colab chỉ hỗ trợ 2 luồng duy nhất).
- `optimizer='AdamW'`: thuật toán AdamW – bản cải tiến của Adam, với trọng số `weight decay` tốt hơn, giúp `generalization` tốt hơn.
- `cos_lr=True`: sử dụng cosine learning rate scheduler làm giảm learning rate (theo hình sin/cos) để tránh việc học quá nhanh hoặc chậm.
- `augment=True`: bật các kỹ thuật tăng cường dữ liệu trong quá trình huấn luyện.

Mô hình được huấn luyện trong tối đa 50 epoch với kích thước batch là 32, sử dụng thuật toán tối ưu AdamW – một biến thể cải tiến của Adam với khả năng điều chỉnh trọng số tốt hơn, giúp mô hình tổng quát hóa tốt hơn trên dữ liệu kiểm tra. Đồng thời, quá trình huấn luyện còn áp dụng lịch trình điều chỉnh tốc độ học theo dạng cosine nhằm giảm dần tốc độ học theo chu kỳ, giúp mô hình học ổn định và tránh hiện tượng overshooting.

Để tăng tính đa dạng của dữ liệu đầu vào và giảm nguy cơ overfitting, các kỹ thuật tăng cường dữ liệu mà YOLO hỗ trợ như lật ảnh, thay đổi độ sáng, mosaic... được kích hoạt trong suốt quá trình huấn luyện. Mô hình được thiết lập cơ chế dừng sớm với ngưỡng kiên nhẫn là 10 epoch – nếu không có



sự cải thiện đáng kể về độ chính xác, quá trình huấn luyện sẽ tự động kết thúc nhằm tiết kiệm thời gian và tài nguyên. Ngoài ra, trọng số của mô hình được lưu lại sau mỗi 10 epoch để đảm bảo khả năng khôi phục nếu xảy ra lỗi giữa chừng. Kết quả huấn luyện bao gồm các biểu đồ loss, chỉ số mAP, các file trọng số (best.pt, last.pt) được lưu tại thư mục /runs/food\_calor\_project.

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size			
36/50	4.54G	0.5285	0.3435	0.8923	20	640: 100%	66/66	[00:40<00:00, 1.63it/s]	
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100%	10/10	[00:05<00:00, 1.69it/s]
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size			
37/50	4.54G	0.5202	0.3395	0.89	20	640: 100%	66/66	[00:39<00:00, 1.65it/s]	
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100%	10/10	[00:07<00:00, 1.34it/s]
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size			
38/50	4.54G	0.52	0.3258	0.8873	12	640: 100%	66/66	[00:39<00:00, 1.65it/s]	
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100%	10/10	[00:06<00:00, 1.60it/s]
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size			
39/50	4.54G	0.522	0.3282	0.8961	17	640: 100%	66/66	[00:40<00:00, 1.63it/s]	
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100%	10/10	[00:06<00:00, 1.54it/s]
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size			
40/50	4.54G	0.5122	0.3213	0.8927	14	640: 100%	66/66	[00:39<00:00, 1.66it/s]	
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100%	10/10	[00:07<00:00, 1.34it/s]
	all	596	1218	0.983	0.987	0.989	0.913		

Hình 3.11 Kết quả huấn luyện mô hình YOLOv8 qua một vài epoch

### 3.4.2. Hồi quy tuyến tính

Mô hình hồi quy tuyến tính được huấn luyện nhằm ước lượng khối lượng (gram) của thực phẩm dựa trên thể tích (cm<sup>3</sup>). Dữ liệu huấn luyện được lấy từ tệp excel mà bộ dữ liệu cung cấp chứa các cặp giá trị thể tích và khối lượng được thu thập thực tế. Để xử lý các định dạng số không thống nhất (dấu phẩy thay vì dấu chấm thập phân), toàn bộ giá trị khối lượng sẽ được chuẩn hóa về dạng số thực (float) trong quá trình đọc dữ liệu từ tệp excel.

Mỗi loại thực phẩm (ứng với mỗi sheet trong file Excel) được xử lý riêng biệt, mô hình hồi quy tuyến tính được huấn luyện độc lập trên từng tập dữ liệu này để tìm ra mối quan hệ tuyến tính tốt nhất giữa thể tích và khối lượng. Hệ số hồi quy (coefficient) và hệ số chặn (intercept) được rút trích từ mô hình với công thức:  $weight = coefficient * volume + intercept$ . Trong đó, weight là khối lượng và volume là thể tích của một món ăn.

```

1 usage
8  def train_model_per_sheet():
9      regression_models = {}
10     for sheet_name, df in xls.items():
11         if 'volume(cm^3)' not in df.columns or 'weight(g)' not in df.columns:
12             print(f"skip sheet {sheet_name}: missing required column.")
13             continue
14         df['weight(g)'] = df['weight(g)'].astype(str).str.replace(',', '.').astype(float)
15         X = df[['volume(cm^3)']].values
16         y = df['weight(g)'].values
17
18         model = LinearRegression()
19         model.fit(X, y)
20
21         a = model.coef_[0]
22         b = model.intercept_
23
24         regression_models[sheet_name] = (round(a, 6), round(b, 6))
25         print(f"{sheet_name}: weight = {a:.4f} * volume + {b:.4f}")
26     return regression_models

```

Hình 3.12 Hàm xử lý dữ liệu và huấn luyện mô hình hồi quy tuyến tính

Thuật toán sử dụng trong huấn luyện là Ordinary Least Squares (OLS) – phương pháp cổ điển và hiệu quả trong việc tìm đường thẳng hồi quy tối ưu nhằm tối thiểu sai số bình phương giữa giá trị thực tế và giá trị dự đoán. Việc huấn luyện diễn ra rất nhanh do dữ liệu đơn giản và mô hình tuyến tính có độ phức tạp thấp.

Các mô hình sau huấn luyện được lưu trữ dưới dạng hệ số hồi quy trong một tệp python regression\_models.py, nhằm phục vụ cho việc tích hợp vào hệ thống ước lượng calo.

```

1 usage
29  def write_weights_to_file(regression_models):
30      with open('../food-calo-estimate-system/models/regression_models.py', 'w') as f:
31          f.write("# Hệ số hồi quy volume → weight theo từng loại thực phẩm\n")
32          f.write("regression_models = {\n")
33          for food_type, (a, b) in regression_models.items():
34              f.write(f"    '{food_type}': ({a}, {b}),\n")
35          f.write("}\n")

```

Hình 3.13 Hàm ghi các hệ số hồi quy ra tệp

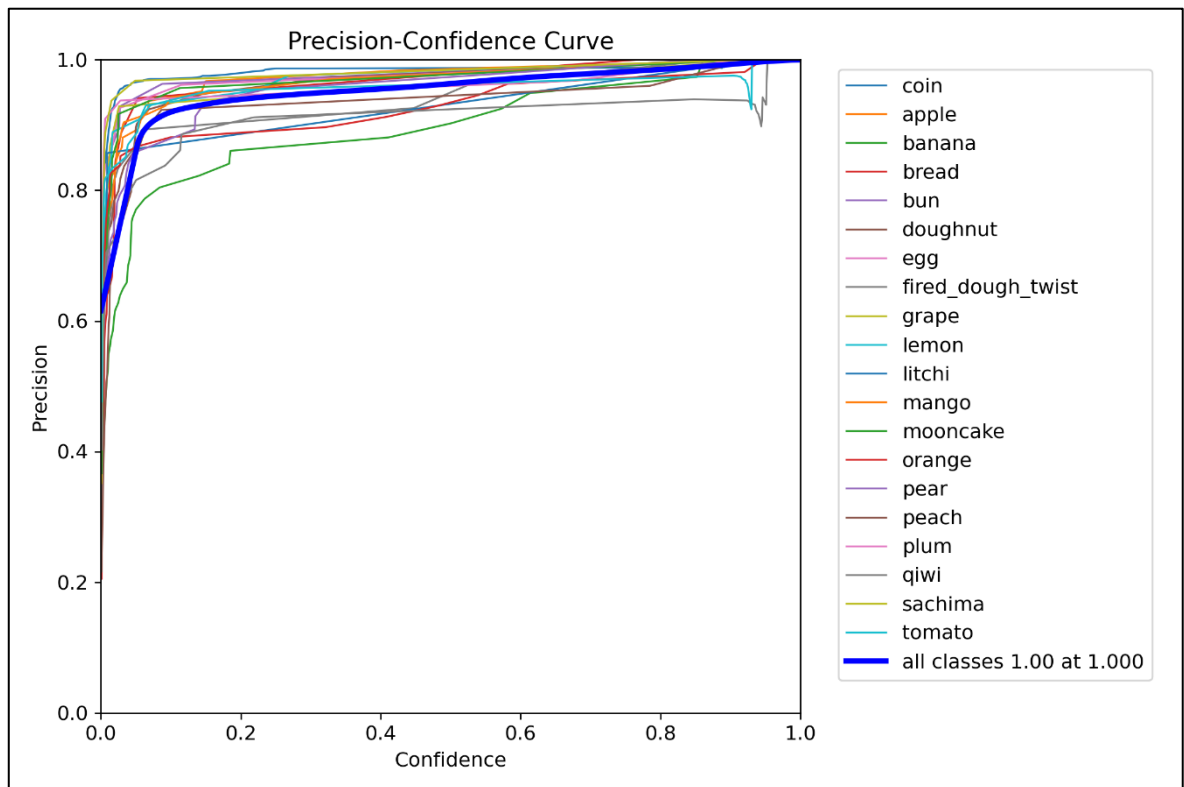
## 3.5. Kết quả thực nghiệm

### 3.5.1. YOLOv8

Qua 50 epoch huấn luyện, ta có thể thấy sự cải thiện rõ rệt về hiệu suất của mô hình. Ban đầu, các chỉ số như độ chính xác (precision), độ bao phủ

(recall), và mAP đều ở mức rất thấp khi mới khởi tạo. Tuy nhiên, theo thời gian các giá trị này tăng lên đáng kể, minh chứng cho việc mô hình đang học tốt và khả năng nhận diện, phân loại ngày càng chính xác hơn. Các giá trị loss (box\_loss, cls\_loss, dfl\_loss) đều giảm dần qua các epoch, cho thấy mô hình ngày càng tối ưu hơn trong việc dự đoán vị trí và phân loại đối tượng. Tổng quan, mô hình có sự tiến bộ tốt và đạt hiệu quả khá cao trong bài toán nhận diện đối tượng.

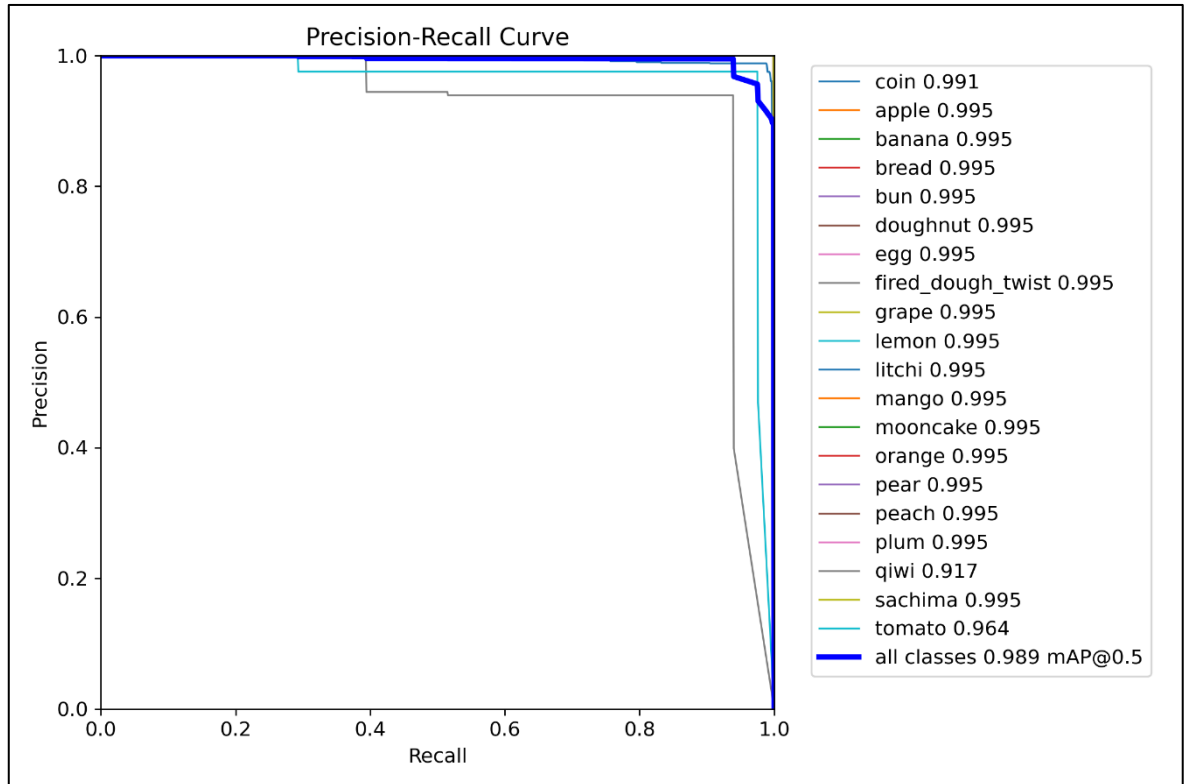
Qua biểu đồ Precision-Confidence Curve cho từng lớp đối tượng trong mô hình, ta nhận thấy rằng: độ chính xác cao khi hầu hết các lớp đều đạt độ chính xác gần 1.0, độ chính xác tăng nhanh và ổn định khi độ tin cậy  $> 0.5$ . Tuy nhiên, có một số đường cong dao động nhẹ ở độ tin cậy thấp do số lượng mẫu còn ít.



Hình 3.14 Biểu đồ Precision-Confidence Curve

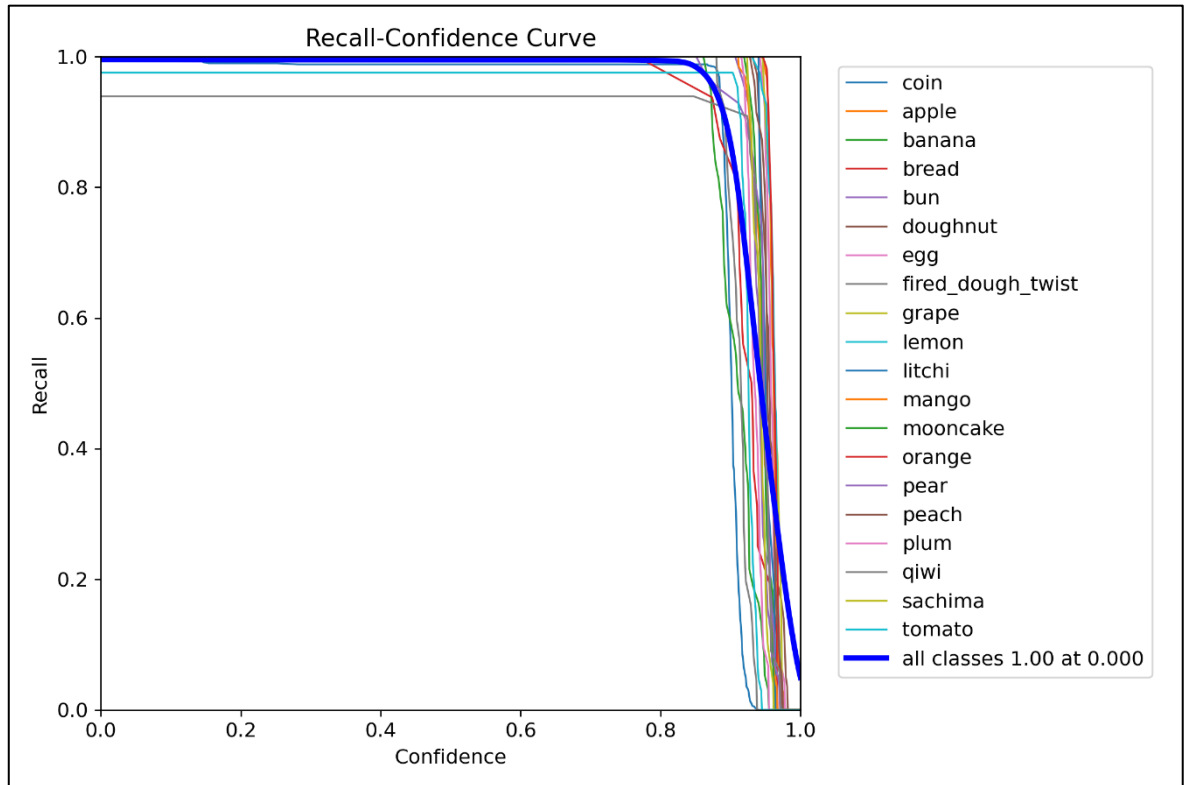
Biểu đồ Precision-Recall Curve cho biết hiệu năng của mô hình: mô hình đưa ra dự đoán đúng và rất ít bỏ sót các đối tượng cần phát hiện. Đường cong lệch về phía góc trên phải (recall = 1.0, precision = 1.0) biểu thị hiệu năng tốt của mô hình trong việc phát hiện đối tượng: hầu hết các giá trị mAP

đều đạt kết quả cao, khoảng 0.995 và giá trị mAP@0.5 đạt 0.989. Bên cạnh đó, nhận thấy độ chính xác của 2 lớp “qiwi” và “tomato” đang thấp hơn hẳn so với các lớp còn lại.



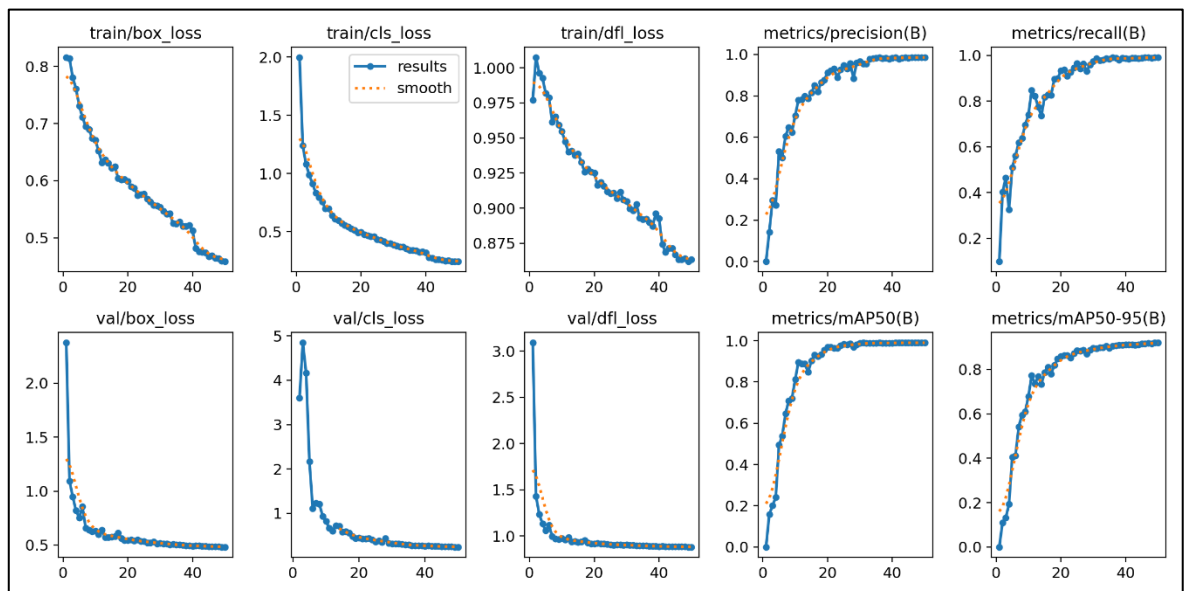
Hình 3.15 Biểu đồ Precision-Recall Curve

Biểu đồ Recall-Confidence Curve cung cấp thông tin giá trị về hiệu năng của mô hình YOLOv8. Với mức độ bao phủ cao trên hầu hết các lớp, mô hình cho thấy khả năng phát hiện đối tượng tốt, đặc biệt ở các ngưỡng tin cậy thấp. Tuy nhiên, hiệu năng cho các lớp có độ bao phủ giảm mạnh như "kiwi" và "tomato" do sự mất cân bằng về dữ liệu.



Hình 3.16 Biểu đồ Recall-Confidence Curve

Dưới đây là hình 3.14 cho biết các kết quả chi tiết của mô hình trên tập huấn luyện và xác thực.



Hình 3.17 Kết quả của mô hình

Trong quá trình huấn luyện, các giá trị loss như train/box\_loss, train/cls\_loss, và train/df\_l\_loss đều giảm dần theo số epoch, điều này thể hiện sự hội tụ tốt của mô hình.

- train/box\_loss giảm từ 0.8 xuống dưới 0.5, chứng tỏ mô hình ngày càng dự đoán chính xác hơn về vị trí và kích thước của hộp giới hạn.
- train/cls\_loss (classification loss) giảm đáng kể từ 2.0 xuống dưới 0.5, điều này cho thấy khả năng phân loại của mô hình được cải thiện rõ rệt.
- train/df\_l\_loss (distribution focal loss) cũng giảm dần, biểu thị việc mô hình tối ưu hóa được độ chính xác trong phân phối xác suất của hộp giới hạn.

Trong quá trình xác thực, các chỉ số tương tự trên tập kiểm thử cũng có xu hướng cải thiện tương đồng với tập huấn luyện. Các giá trị loss như val/box\_loss, val/cls\_loss, và val/df\_l\_loss giảm đáng kể theo số epoch, với val/box\_loss đạt mức dưới 0.5 và val/cls\_loss từ 5 giảm xuống gần 1.

Ngoài ra, các giá trị metrics/precision(B) và metrics/recall(B) liên tục tăng qua từng epoch, lần lượt đạt giá trị tiệm cận 1.0, minh chứng rằng mô hình có khả năng phát hiện đối tượng với độ chính xác và độ bao phủ cao. Các chỉ số hiệu suất như metrics/mAP50(B) và metrics/mAP50-95(B) tăng dần và đạt giá trị gần 0.9 ở cuối quá trình huấn luyện. Điều này chứng tỏ mô hình không chỉ học tốt trên tập huấn luyện mà còn có khả năng tổng quát hóa tốt.

Các chỉ số hiệu suất thu được khi huấn luyện mô hình được mô tả trong bảng dưới.

*Bảng 3.1 Bảng hiệu suất sau khi huấn luyện mô hình*

<b>Chỉ số</b> <b>Model</b>	<b>Precision</b>	<b>Recall</b>	<b>mAP@0.5</b>	<b>mAP@0.5:0.95</b>
last.pt	0.986	0.992	0.99	0.921
best.pt	0.987	0.994	0.989	0.914

Nhận xét:

- Bảng trên thể hiện các chỉ số đánh giá của mô hình YOLOv8 trên tập xác thực, bao gồm hai phiên bản mô hình: last.pt (trọng số tại epoch

cuối cùng) và best.pt (trọng số có hiệu suất tốt nhất trong quá trình huấn luyện, dựa theo chỉ số  $mAP@0.5:0.95$  cao nhất).

- Kết quả cho thấy cả hai mô hình đều đạt hiệu suất rất cao với các chỉ số Precision, Recall,  $mAP@0.5$  và  $mAP@0.5:0.95$  đều vượt ngưỡng 0.9. Cụ thể, mô hình best.pt đạt Precision là 0.987, Recall là 0.994,  $mAP@0.5$  là 0.989 và  $mAP@0.5:0.95$  là 0.914. Trong khi đó, mô hình last.pt có chỉ số Precision và Recall thấp hơn một chút, lần lượt là 0.986 và 0.992, nhưng lại nhỉnh hơn ở  $mAP@0.5$  (0.99) và  $mAP@0.5:0.95$  (0.921).
- Sự chênh lệch giữa các chỉ số là không đáng kể, cho thấy mô hình có tính ổn định cao trong suốt quá trình huấn luyện. Trong báo cáo, mô hình last.pt sẽ được chọn để sử dụng trong việc triển khai tiếp theo vì độ chính xác  $mAP@0.5:0.95$  cao nhất.

### 3.5.2. Hồi quy tuyến tính

Kết quả huấn luyện cho thấy các hệ số hồi quy có sự khác biệt rõ rệt giữa các loại thực phẩm, phản ánh mật độ và cấu trúc vật lý riêng của từng loại.

```

8  # Hệ số hồi quy volume → weight theo từng loại thực phẩm
9  # Mô hình: weight = coefficient * volume + intercept
10 regression_models = {
11     'apple': (0.689415, 30.876216),
12     'banana': (0.626376, 48.26164),
13     'bread': (0.006667, 26.875),
14     'bun': (0.194086, 35.21191),
15     'doughnut': (0.173122, 25.88878),
16     'egg': (1.07, 4.7),
17     'fired_dough_twist': (0.159125, 30.55875),
18     'grape': (1.088095, -41.642857),
19     'lemon': (0.163333, 77.6),
20     'litchi': (1.1975, -8.375),
21     'mango': (0.978617, 10.01383),
22     'mooncake': (0.734314, 13.257843),
23     'orange': (0.399855, 110.877126),
24     'pear': (0.943484, 1.907932),
25     'peach': (1.073654, -5.769231),
26     'plum': (1.03, -2.7),
27     'qiwi': (0.907095, 9.873311),
28     'sachima': (0.2725, -8.3),
29     'tomato': (0.666538, 58.815385),
30 }

```

*Hình 3.18 Kết quả huấn luyện mô hình hồi quy tuyến tính*

Toàn bộ hệ số được lưu trữ trong một tệp python `regression_models.py`, giúp hệ thống có thể nhanh chóng truy xuất và sử dụng các mô hình hồi quy này cho mục tiêu ước lượng khối lượng và từ đó ước lượng lượng calo của từng loại thực phẩm thông qua thể tích đã đo được.

Mô hình tuyến tính tuy đơn giản nhưng cho phép tích hợp hiệu quả vào pipeline suy luận thời gian thực, nhờ vào khả năng tính toán nhanh và chi phí triển khai thấp.



## CHƯƠNG 4. XÂY DỰNG HỆ THỐNG

### 4.1. Giới thiệu một số công nghệ sử dụng

#### 4.1.1. Web framework

Flask là một micro web framework được phát triển bằng ngôn ngữ lập trình Python, hướng đến việc đơn giản hóa quy trình xây dựng các ứng dụng web. Dù có thiết kế tối giản, Flask vẫn cung cấp đầy đủ các công cụ cần thiết để phát triển ứng dụng, từ các dự án cơ bản đến các hệ thống có quy mô trung bình.

Điểm mạnh nổi bật của Flask là tính linh hoạt và khả năng mở rộng cao. Flask không áp đặt kiến trúc cứng nhắc, cho phép lập trình viên tự do lựa chọn và tích hợp các thư viện hoặc mô-đun bên ngoài theo nhu cầu. Điều này đặc biệt phù hợp với các dự án vừa và nhỏ, các ứng dụng thử nghiệm ý tưởng mới, hoặc các hệ thống cần phản hồi nhanh mà không cần cấu trúc phức tạp. Flask cung cấp nhiều tính năng nổi bật hỗ trợ quá trình phát triển:

- Máy chủ phát triển tích hợp sẵn: Giúp kiểm thử và triển khai nhanh trong môi trường nội bộ.
- Trình gỡ lỗi mạnh mẽ: Hỗ trợ phát hiện và xử lý lỗi trong quá trình phát triển.
- Tương thích tốt với Google App Engine: Dễ dàng triển khai lên các nền tảng đám mây của Google.
- Hệ sinh thái mở rộng phong phú: Bao gồm các tiện ích như xác thực người dùng, ORM, RESTful API, v.v.
- Hỗ trợ bảo mật và xử lý HTTP: Tích hợp các cơ chế bảo vệ cookie, chống CSRF, xử lý request hiệu quả.
- Triển khai RESTful API dễ dàng: Giao diện lập trình rõ ràng, thuận tiện cho việc xây dựng API.

Nhờ vào sự tối giản nhưng hiệu quả, Flask được xem là lựa chọn phù hợp để phát triển hệ thống, giúp giảm thời gian triển khai và tăng tính linh hoạt khi mở rộng về sau.

#### 4.1.2. Hệ quản trị cơ sở dữ liệu

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL phổ biến, được thiết kế để lưu trữ dữ liệu dưới dạng document (tài liệu) theo cấu trúc JSON. Khác với các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) sử dụng bảng và cột, MongoDB tổ chức dữ liệu thành các collection và document, mang lại sự linh hoạt cao trong lưu trữ và truy vấn.

Một trong những điểm mạnh của MongoDB là khả năng xử lý dữ liệu phi cấu trúc hoặc bán cấu trúc, cho phép hệ thống dễ dàng mở rộng và thay đổi mô hình dữ liệu mà không cần phải thiết kế lại toàn bộ cấu trúc cơ sở dữ liệu. Điều này đặc biệt hữu ích trong các ứng dụng hiện đại có dữ liệu phong phú, thay đổi thường xuyên, chẳng hạn như ảnh, metadata, hoặc kết quả phân tích từ mô hình AI.

Trong hệ thống được xây dựng, MongoDB được sử dụng để lưu trữ các thông tin liên quan đến ảnh món ăn, kết quả nhận diện từ mô hình YOLOv8, cũng như dữ liệu người dùng và lịch sử truy vấn. Sự linh hoạt của MongoDB giúp dễ dàng thao tác, cập nhật và truy vấn dữ liệu phục vụ cho việc hiển thị kết quả cũng như huấn luyện mô hình ước lượng calo. Một số đặc điểm nổi bật của MongoDB:

- Lưu trữ linh hoạt: Hỗ trợ lưu dữ liệu dưới dạng document với cấu trúc động, không cần schema cố định.
- Tốc độ truy xuất nhanh: Nhờ cơ chế indexing và tối ưu lưu trữ, MongoDB cho phép truy vấn dữ liệu hiệu quả.
- Khả năng mở rộng tốt: Dễ dàng mở rộng theo chiều ngang (horizontal scaling) để xử lý khối lượng dữ liệu lớn.
- Hỗ trợ tích hợp mạnh với Python: Thông qua thư viện pymongo, MongoDB dễ dàng kết nối và thao tác từ ứng dụng Flask.

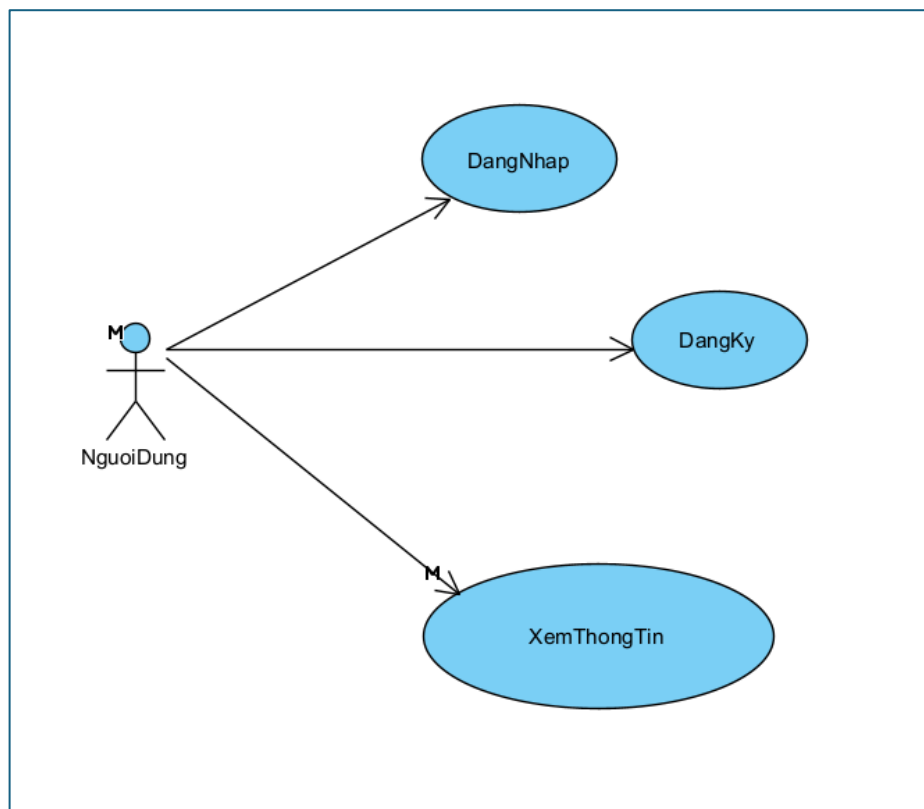
- Hỗ trợ thao tác JSON tự nhiên: Phù hợp với cách biểu diễn dữ liệu trong các hệ thống AI và web hiện đại.

Nhờ những ưu điểm trên, MongoDB là lựa chọn phù hợp để lưu trữ dữ liệu động và hỗ trợ hiệu quả cho các thao tác xử lý ảnh và thông tin đầu ra của mô hình.

## 4.2. Phân tích thiết kế hệ thống

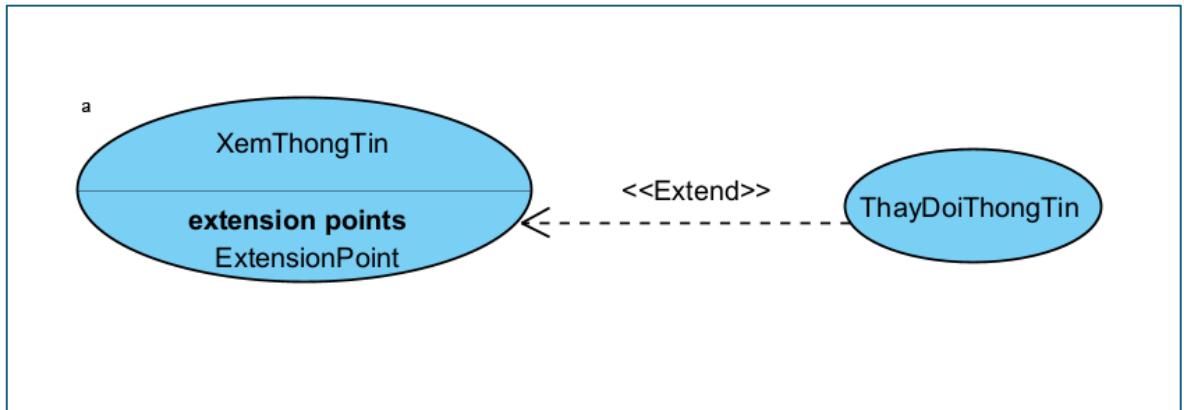
### 4.2.1. Biểu đồ use case

**Biểu đồ use case tổng quát:**



*Hình 4.1 Biểu đồ use case tổng quát*

**Phân rã use case Xem thông tin:**



Hình 4.2 Biểu đồ phân rã use case Xem thông tin

#### 4.2.2. Đặc tả use case

##### Đặc tả use case Đăng ký

Bảng 4.1 Bảng mô tả use case Đăng ký

<b>Mã use case</b>	UC1
<b>Tên use case</b>	Đăng ký
<b>Tóm tắt</b>	Use case này cho phép người dùng đăng ký tài khoản
<b>Actor</b>	Người dùng
<b>Tiền điều kiện</b>	Người dùng đang truy cập vào giao diện đăng ký
<b>Đảm bảo tối thiểu</b>	Hệ thống hiển thị thông báo lỗi nếu người dùng nhập thiếu hoặc sai định dạng thông tin. Không tạo tài khoản nếu thông tin không hợp lệ.
<b>Đảm bảo thành công</b>	Một tài khoản mới được tạo thành công trong hệ thống, Người dùng được chuyển hướng đến trang đăng nhập hoặc trang chủ.
<b>Kích hoạt</b>	Khi người dùng truy cập vào trang đăng ký và chọn chức năng đăng ký.
<b>Luồng sự kiện:</b>	
<ul style="list-style-type: none"> <li>- Luồng cơ bản:</li> <li>1) Người dùng truy cập vào trang đăng ký.</li> <li>2) Hệ thống hiển thị biểu mẫu đăng ký.</li> <li>3) Người dùng nhập các thông tin yêu cầu: họ tên, email, mật khẩu,...</li> </ul>	

4) Người dùng nhấn nút “Đăng ký”. 5) Hệ thống kiểm tra tính hợp lệ của các thông tin đã nhập. 6) Nếu thông tin hợp lệ, hệ thống lưu thông tin vào cơ sở dữ liệu và tạo tài khoản mới. 7) Hệ thống thông báo đăng ký thành công và chuyển hướng người dùng.
<b>Ngoại lệ:</b> Không có
<b>Hậu điều kiện:</b> Không có

### Đặc tả use case Đăng nhập:

*Bảng 4.2 Bảng mô tả use case Đăng nhập*

<b>Mã use case</b>	UC2
<b>Tên use case</b>	Đăng nhập
<b>Tóm tắt</b>	Use case này cho phép người dùng đăng nhập vào hệ thống bằng thông tin tài khoản đã đăng ký
<b>Actor</b>	Người dùng
<b>Tiền điều kiện</b>	Người dùng đã có tài khoản hợp lệ trong hệ thống. Người dùng đang truy cập giao diện đăng nhập.
<b>Đảm bảo tối thiểu</b>	Hệ thống không cho phép truy cập nếu thông tin đăng nhập sai. Hiện thị thông báo lỗi khi thông tin không hợp lệ.
<b>Đảm bảo thành công</b>	Người dùng được xác thực và đăng nhập thành công. Hệ thống chuyển hướng người dùng đến trang chính hoặc trang cá nhân.
<b>Kích hoạt</b>	Người dùng chọn chức năng “Đăng nhập” từ trang chủ.
<b>Luồng sự kiện:</b> <ul style="list-style-type: none"> <li>- Luồng cơ bản:             <ol style="list-style-type: none"> <li>1) Người dùng truy cập trang đăng nhập.</li> <li>2) Hệ thống hiển thị biểu mẫu đăng nhập.</li> <li>3) Người dùng nhập email và mật khẩu.</li> </ol> </li> </ul>	

4) Người dùng nhấn nút “Đăng nhập”. 5) Hệ thống kiểm tra thông tin đăng nhập. 6) Nếu thông tin đúng, hệ thống xác thực tài khoản. Hệ thống thông báo đăng nhập thành công và chuyển hướng người dùng.
<b>Ngoại lệ:</b> Không có
<b>Hậu điều kiện:</b> Không có

**Đặc tả use case Xem thông tin:**

*Bảng 4.3 Bảng mô tả use case Xem thông tin*

<b>Mã use case</b>	UC3
<b>Tên use case</b>	Xem thông tin
<b>Tóm tắt</b>	Use case này cho phép người dùng xem thông tin cá nhân của mình và xem lịch sử các lần đã sử dụng chức năng phân tích hình ảnh để ước lượng calo.
<b>Actor</b>	Người dùng
<b>Tiền điều kiện</b>	Người dùng đã đăng nhập vào hệ thống
<b>Đảm bảo tối thiểu</b>	Hệ thống hiển thị được thông tin cá nhân cơ bản và lịch sử sử dụng hệ thống ước lượng calo.
<b>Đảm bảo thành công</b>	Thông tin cá nhân và danh sách các kết quả phân tích ước lượng calo được hiển thị chính xác cho người dùng.
<b>Kích hoạt</b>	Người dùng chọn chức năng “Xem thông tin” từ giao diện trang chủ.
<b>Luồng sự kiện:</b> <ul style="list-style-type: none"> <li>- Luồng cơ bản:               <ol style="list-style-type: none"> <li>1) Người dùng nhấn vào mục “Xem thông tin” trên thanh menu.</li> <li>2) Hệ thống truy xuất thông tin cá nhân từ cơ sở dữ liệu.</li> <li>3) Hệ thống truy xuất lịch sử các lần phân tích hình ảnh: thời gian, hình ảnh, kết quả calo.</li> <li>4) Hệ thống hiển thị thông tin cá nhân và bảng lịch sử phân tích calo.</li> </ol> </li> </ul>	

<b>Ngoại lệ:</b> Không có
<b>Hậu điều kiện:</b> Không có

**Đặc tả use case Thay đổi thông tin:**

*Bảng 4.4 Bảng mô tả use case Thay đổi thông tin*

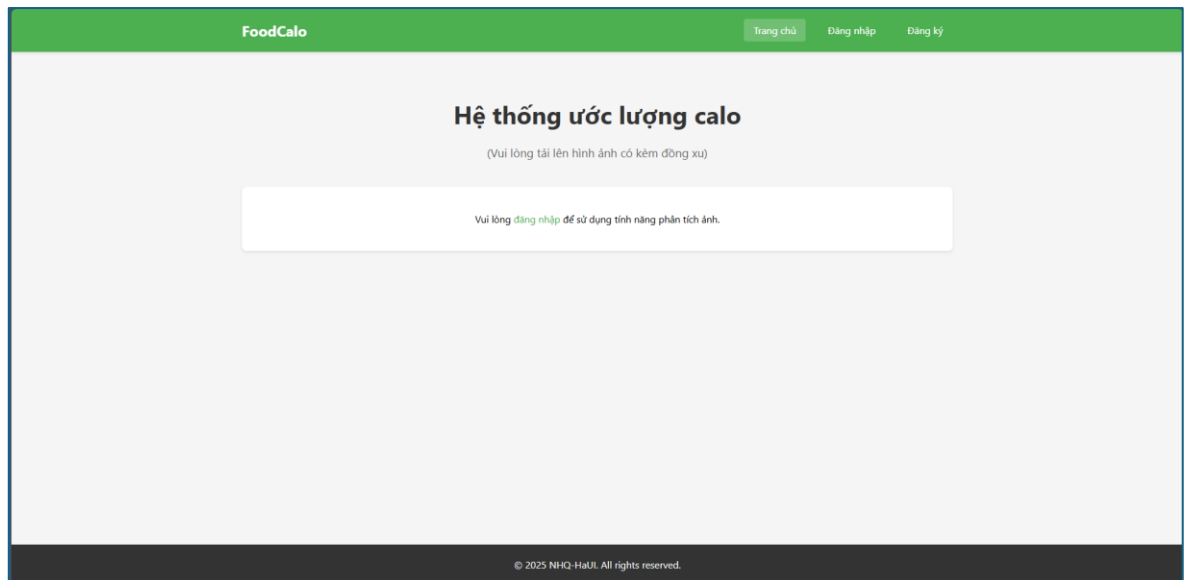
<b>Mã use case</b>	UC4
<b>Tên use case</b>	Thay đổi thông tin
<b>Tóm tắt</b>	Use case này cho phép người dùng cập nhật thông tin cá nhân
<b>Actor</b>	Người dùng
<b>Tiền điều kiện</b>	Người dùng đã đăng nhập vào hệ thống
<b>Đảm bảo tối thiểu</b>	Hệ thống không lưu thông tin nếu dữ liệu không hợp lệ. Hiện thị thông báo lỗi tương ứng khi nhập sai.
<b>Đảm bảo thành công</b>	Thông tin người dùng được cập nhật chính xác trong hệ thống. Hiện thị thông báo cập nhật thành công.
<b>Kích hoạt</b>	Người dùng chọn chức năng “Thay đổi thông tin” từ giao diện xem thông tin.
<b>Luồng sự kiện:</b> <ul style="list-style-type: none"> <li>- Luồng cơ bản:</li> <li>1) Người dùng chọn mục “Thay đổi thông tin”.</li> <li>2) Hệ thống truy xuất và hiển thị thông tin cá nhân hiện tại trong biểu mẫu.</li> <li>3) Người dùng cập nhật các trường mong muốn: tên, mật khẩu.</li> <li>4) Người dùng nhấn nút “Lưu thay đổi”.</li> <li>5) Hệ thống kiểm tra tính hợp lệ của thông tin mới.</li> <li>6) Nếu hợp lệ, hệ thống cập nhật thông tin trong cơ sở dữ liệu. Hệ thống hiển thị thông báo cập nhật thành công.</li> </ul>	
<b>Ngoại lệ:</b> Không có	
<b>Hậu điều kiện:</b> Không có	

### 4.3. Một số kết quả

Hệ thống được xây dựng với chức năng chính là ước lượng lượng calo trong bữa ăn dựa trên hình ảnh được người dùng tải lên. Ngay sau khi hoàn tất phân tích, hệ thống sẽ cung cấp thông tin chi tiết về giá trị dinh dưỡng, bao gồm lượng calo của toàn bộ bữa ăn. Đồng thời, kết quả này có thể được lưu trữ trong lịch sử hỗ trợ mọi người trong việc quản lý chế độ ăn uống và duy trì sức khỏe. Hệ thống không chỉ giúp tiết kiệm thời gian so với các phương pháp ước lượng thủ công mà còn đảm bảo tính chính xác và tiện lợi, phù hợp với nhu cầu theo dõi dinh dưỡng trong lối sống hiện đại.

Giao diện hệ thống được thiết kế theo nguyên tắc đơn giản, dễ sử dụng và thân thiện với người dùng, nhằm hỗ trợ hiệu quả quá trình tương tác giữa người dùng và hệ thống. Hệ thống gồm hai giao diện chính: giao diện đăng nhập và giao diện phân tích calo trong bữa ăn.

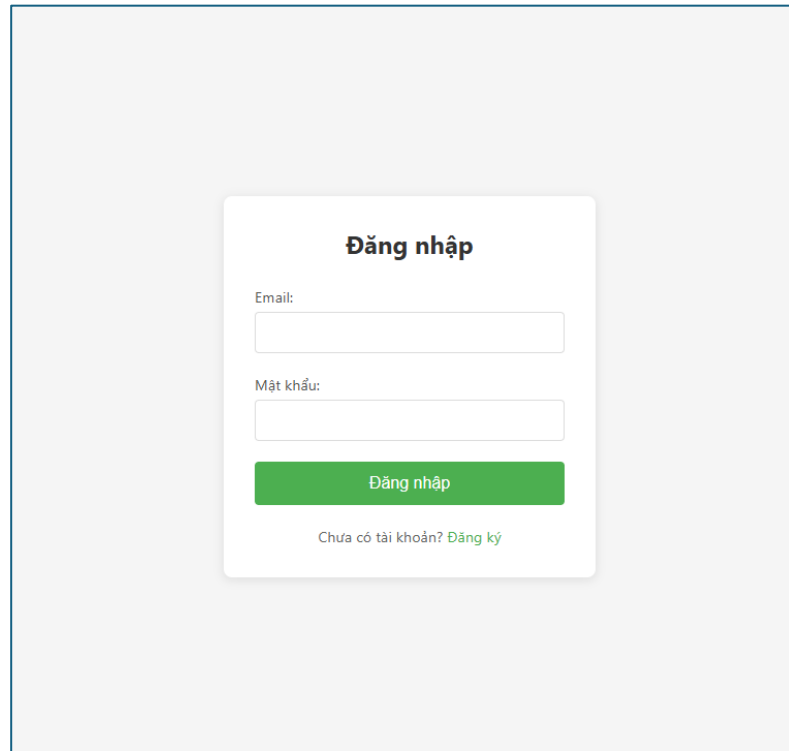
Giao diện khi người dùng mở phần mềm lên được hiển thị như hình 4.1 dưới đây.



*Hình 4.3 Giao diện trang chủ khi người dùng truy cập*

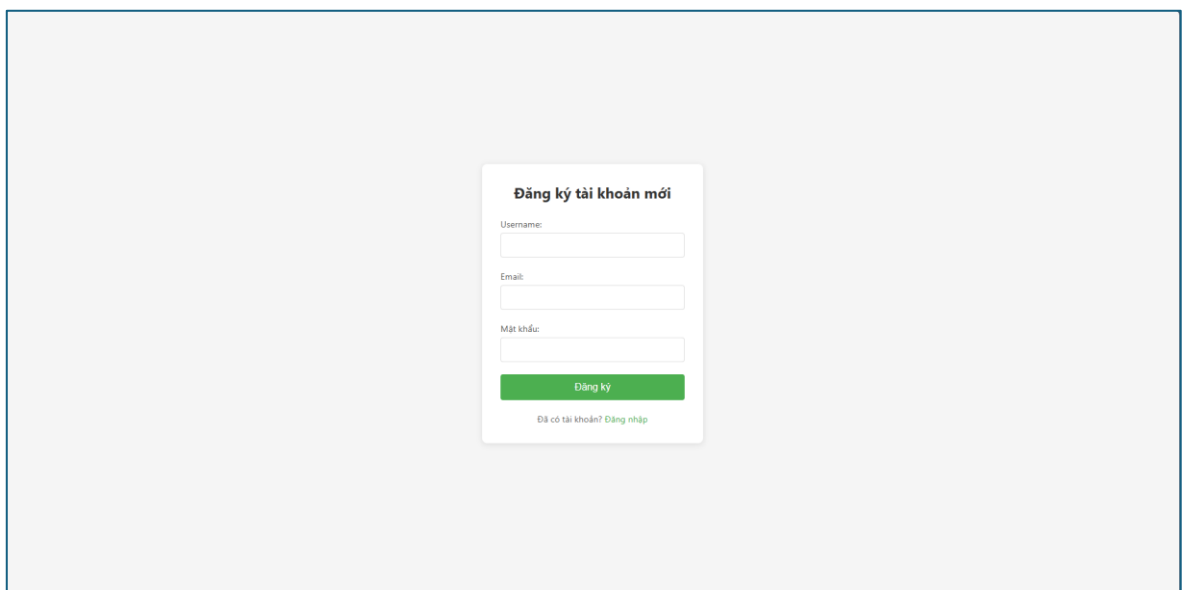
Giao diện người dùng đơn giản gồm chức năng chính là phân tích lượng calo trong bữa ăn. Để sử dụng chức năng phân tích lượng calo trong bữa ăn, người dùng cần đăng nhập tài khoản của mình vào hệ thống.



A login form titled "Đăng nhập" (Login) centered on a light gray background. The form is white with a thin gray border. It contains two input fields: "Email:" and "Mật khẩu:" (Password:). Below the password field is a green button labeled "Đăng nhập". At the bottom, there is a link that says "Chưa có tài khoản? Đăng ký" (Don't have an account? Register).

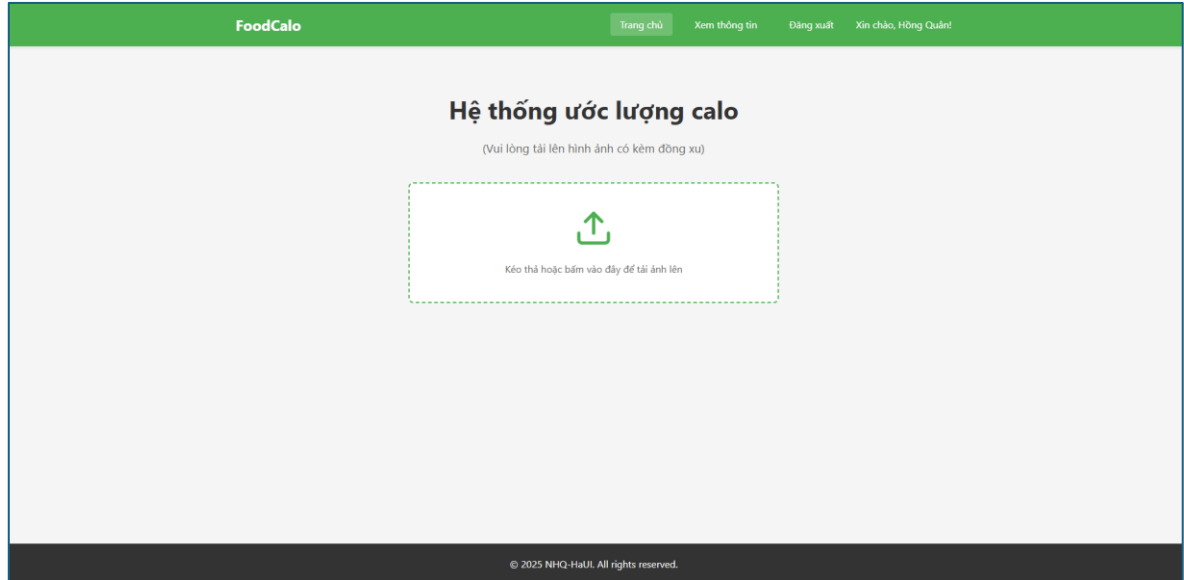
*Hình 4.4 Giao diện màn hình đăng nhập*

Người dùng nhập đầy đủ thông tin tài khoản đã đăng ký vào biểu mẫu đăng nhập trên giao diện. Nếu người dùng không có tài khoản trong hệ thống, người dùng có thể chọn nút “Đăng ký” trên giao diện trang chủ hoặc trong giao diện đăng nhập. Sau khi thực hiện, giao diện đăng ký tài khoản sẽ xuất hiện.

A registration form titled "Đăng ký tài khoản mới" (Register new account) centered on a light gray background. The form is white with a thin gray border. It contains three input fields: "Username:", "Email:", and "Mật khẩu:" (Password:). Below the password field is a green button labeled "Đăng ký". At the bottom, there is a link that says "Đã có tài khoản? Đăng nhập" (Already have an account? Login).

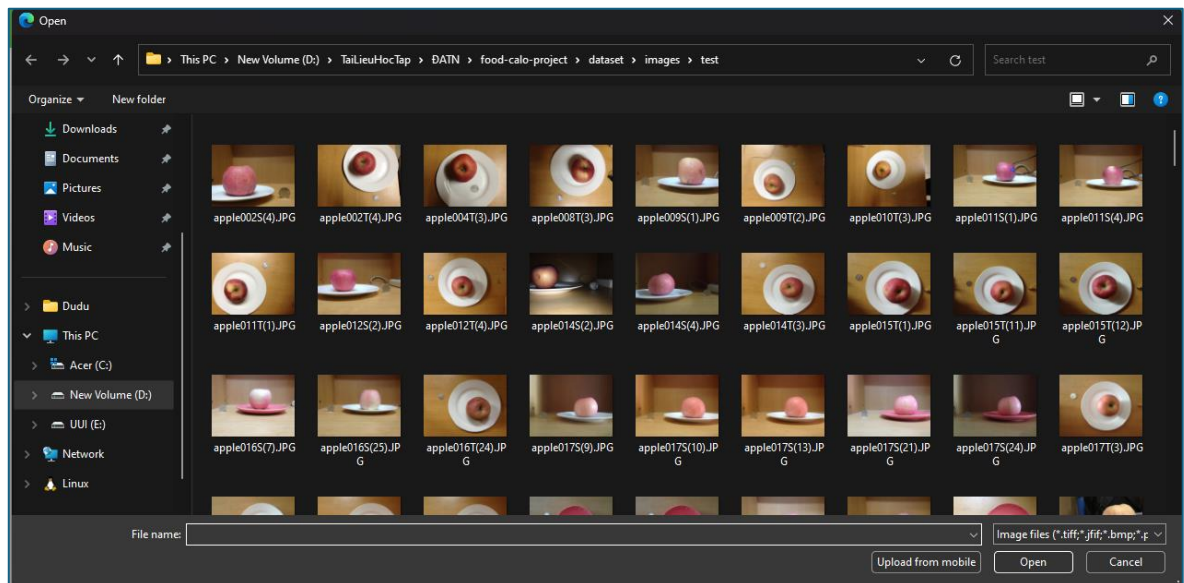
*Hình 4.5 Giao diện màn hình đăng ký*

Người dùng nhập đúng định dạng email, điền đầy đủ thông tin vào biểu mẫu đăng ký và bấm nút “Đăng ký”. Hệ thống sẽ chuyển hướng và đưa người dùng tới giao diện trang chủ sau khi đã đăng nhập.



Hình 4.6 Giao diện người dùng sau khi đã đăng nhập

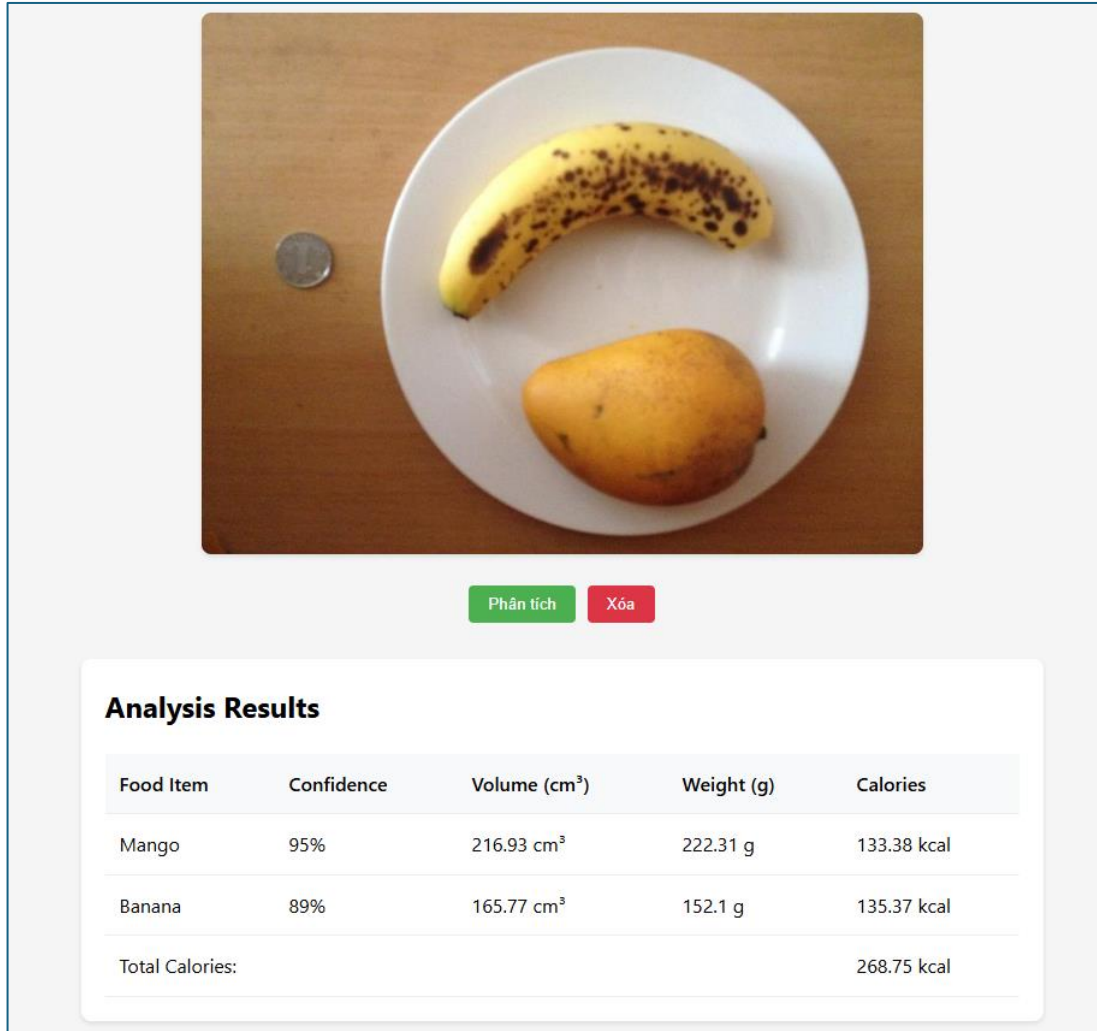
Sau khi đã đăng nhập vào hệ thống, người dùng đã có thể sử dụng chức năng phân tích lượng calo trong bữa ăn. Để sử dụng chức năng, người dùng có thể kéo thả ảnh hoặc bấm chuột vào vùng tải ảnh. Giao diện tải ảnh lên sẽ xuất hiện.



Hình 4.7 Giao diện tải hình ảnh lên

Người dùng chọn ảnh cần phân tích calo và bấm “Open”. Sau khi tải ảnh lên thành công, người dùng thực hiện nhấn nút “Phân tích” để tiến hành

phân tích, ước lượng calo trong hình ảnh. Ở đây sẽ có một ngoại lệ xảy ra khi người dùng tải ảnh lên mà không có vật tham chiếu là đồng xu, hệ thống sẽ báo lỗi và yêu cầu người dùng tải ảnh mới lên.



Phân tích Xóa

**Analysis Results**

Food Item	Confidence	Volume (cm <sup>3</sup> )	Weight (g)	Calories
Mango	95%	216.93 cm <sup>3</sup>	222.31 g	133.38 kcal
Banana	89%	165.77 cm <sup>3</sup>	152.1 g	135.37 kcal
Total Calories:				268.75 kcal

Hình 4.8 Giao diện màn hình ước lượng calo

Bảng kết quả phân tích sẽ hiển thị trên giao diện. Bảng kết quả phân tích bao gồm các thông tin về: món ăn (Food Item), độ tin cậy (Confidence), thể tích món ăn (Volume) tính theo đơn vị cm<sup>3</sup>, khối lượng món ăn (Weight) tính theo đơn vị gam (g), lượng calo trên từng món (Calories) và tổng lượng calo trong bữa ăn (Total Calories).

Người dùng bấm vào “Xem thông tin” để có thể xem thông tin tài khoản, thay đổi thông tin và xem lịch sử ước lượng calo.

FoodCalo				
		Trang chủ	Xem thông tin	Đăng xuất Xin chào, Hồng Quân!
<b>Thông tin cá nhân</b> <b>Tên người dùng:</b> Hồng Quân <b>Email:</b> nhquan@gmail.com <b>Ngày tham gia:</b> 25/05/2025				
Thay đổi thông tin				
<b>Lịch sử ước lượng Calo</b>				
Thời gian	Thực phẩm	Khối lượng (g)	Calories (kcal)	Tổng Calories
25/05/2025 10:54	orange	187.00	87.89	108.27
	tomato	113.21	20.38	
25/05/2025 10:44	mango	222.31	133.38	268.75
	banana	152.10	135.37	
25/05/2025 10:44	mango	222.31	133.38	268.75
	banana	152.10	135.37	
25/05/2025 10:41	mango	222.31	133.38	268.75
	banana	152.10	135.37	

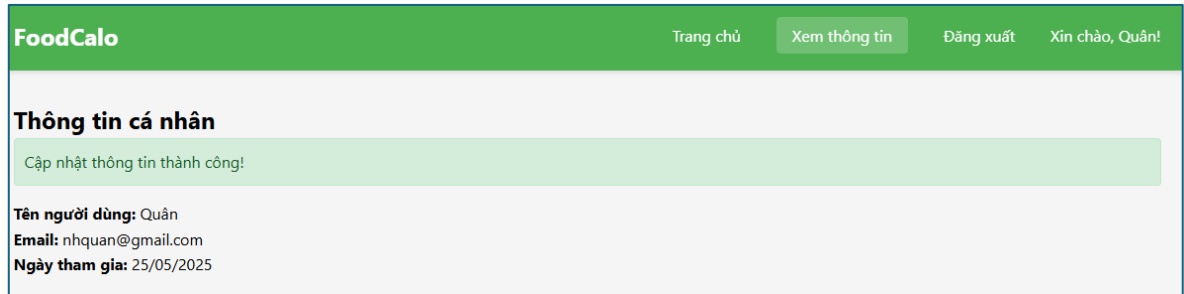
Hình 4.9 Giao diện xem thông tin

Thông tin của người dùng và lịch sử sử dụng chức năng phân tích calo trong bữa ăn sẽ hiển thị lên màn hình. Ở giao diện xem thông tin, người dùng có thể thay đổi thông tin bằng cách nhấn nút “Thay đổi thông tin” trong giao diện xem thông tin. Một biểu mẫu thay đổi thông tin người dùng sẽ xuất hiện.

FoodCalo				
		Trang chủ	Xem thông tin	Đăng xuất Xin chào, Hồng Quân!
<b>Thông tin cá nhân</b> <b>Tên người dùng:</b> Hồng Quân <b>Email:</b> nhquan@gmail.com <b>Ngày tham gia:</b> 25/05/2025				
Thay đổi thông tin				
Tên hiển thị: <input type="text" value="Hồng Quân"/>				
Mật khẩu mới: <input type="password"/>				
Xác nhận mật khẩu: <input type="password"/>				
<input type="button" value="Lưu thay đổi"/> <input type="button" value="Hủy"/>				

Hình 4.10 Giao diện thay đổi thông tin

Người dùng nhập thông tin cần thay đổi và bấm “Lưu thay đổi” để xác nhận thay đổi thông tin. Màn hình sẽ cập nhật thông báo “Cập nhật thông tin thành công!”.



*Hình 4.11 Giao diện thông báo thành công thay đổi thông tin*

## KẾT LUẬN

Trong đồ án này, em đã phát triển một hệ thống hỗ trợ ước lượng lượng calo trong bữa ăn dựa trên hình ảnh do người dùng cung cấp. Hệ thống sử dụng mô hình YOLOv8 để phát hiện và phân loại thực phẩm trong ảnh, từ đó kết hợp với cơ sở dữ liệu dinh dưỡng để tính toán lượng calo và thông tin dinh dưỡng chi tiết. Quy trình của hệ thống bao gồm: xử lý trước ảnh đầu vào, nhận diện thực phẩm, xác định khẩu phần, ước lượng khối lượng khẩu phần và ước lượng calo dựa trên dữ liệu dinh dưỡng đã được tích hợp.

Bên cạnh việc nghiên cứu lý thuyết và thực nghiệm, em đã triển khai một hệ thống cho phép người dùng tải ảnh món ăn lên và nhận được kết quả bao gồm danh sách thực phẩm, lượng calo ước lượng. Ứng dụng này là minh chứng thực tế cho khả năng triển khai của hệ thống, giúp người dùng dễ dàng theo dõi và quản lý chế độ ăn uống.

Tuy nhiên, hệ thống hiện còn một số hạn chế nhất định. Thứ nhất, độ chính xác của mô hình có thể bị ảnh hưởng bởi chất lượng hình ảnh đầu vào, đặc biệt là khi ảnh bị mờ hoặc có nhiều vật thể không liên quan. Thứ hai, cơ sở dữ liệu dinh dưỡng chưa đầy đủ, chưa bao quát được các món ăn đặc thù ở nhiều vùng miền khác nhau. Cuối cùng, sẽ không thể ước lượng được calo chính xác trong bữa ăn của người dùng khi ảnh tải lên không có các vật tham chiếu như trong bộ dữ liệu: đĩa và đồng xu.

Trong thời gian tới, em dự định tiếp tục cải tiến hệ thống theo các hướng sau: nâng cao hiệu quả nhận diện thông qua việc tích hợp các kỹ thuật tiên tiến như attention hoặc thử nghiệm với các mô hình hiện đại như vision transformer để tăng cường độ chính xác và khả năng nhận diện các thực phẩm phức tạp; mở rộng dữ liệu nhờ việc thu thập và xây dựng một cơ sở dữ liệu phong phú hơn, bao gồm các loại thực phẩm đa dạng từ nhiều nền văn hóa và vùng miền khác nhau, giúp hệ thống trở nên toàn diện hơn; phát triển ứng dụng di động có thể triển khai trên nền tảng Android và iOS, cho phép người

dùng dễ dàng sử dụng hệ thống trên thiết bị di động cá nhân, đáp ứng nhu cầu thực tế.

Em hy vọng rằng hệ thống ước lượng calo trong bữa ăn sẽ góp phần hỗ trợ người dùng trong việc kiểm soát chế độ dinh dưỡng, cải thiện sức khỏe và nâng cao chất lượng cuộc sống, đồng thời mở ra tiềm năng phát triển các ứng dụng thông minh hơn trong lĩnh vực chăm sóc sức khỏe.

## TÀI LIỆU THAM KHẢO

- [1] Calo Là Gì? Tìm Hiểu Về Đơn Vị Đo Năng Lượng Của Cơ Thể. URL: <https://mdbuddy.vn/calo-la-gi/> [Đã truy cập 22 03 2025].
- [2] T. Kudo, Báo cáo xu hướng ăn uống lành mạnh tại Việt Nam. URL: <https://www.fhmvietnam.com/tin-tuc-thi-truong/bao-caoxu-huong-an-uong-lanh-manhtai-viet-nam-moi-nhat> (Truy cập lần cuối 22/03/2025).
- [3] A. Vina, AI trong dinh dưỡng: Tối ưu hóa chế độ ăn uống lành mạnh bằng công nghệ thị giác máy tính. URL: <https://www.ultralytics.com/vi/blog/ai-in-nutrition-streamlining-healthy-eating-with-computer-vision> (Truy cập lần cuối 22/03/2025).
- [4] URL: [https://www.pngitem.com/middle/hRJRTm\\_deep-learning-ai-could-deep-learning-come-to/](https://www.pngitem.com/middle/hRJRTm_deep-learning-ai-could-deep-learning-come-to/) (Truy cập lần cuối 23/03/2025).
- [5] Nguyễn Thị Phương Nga và Trần Hùng Cường, Giáo trình trí tuệ nhân tạo, Nhà xuất bản Thống kê, 2021.
- [6] Nguyễn Thị Quỳnh Trang, Convolutional Neural Network là gì? Tìm hiểu về mô hình mạng. URL: <https://lptech.asia/kien-thuc/convolutional-neural-network-la-gi-tim-hieu-ve-thuat-toan-cnn> (Truy cập lần cuối 23/03/2025).
- [7] VinBigData, Giới thiệu một số thuật toán phát hiện đối tượng phổ biến. URL: <https://vinbigdata.com/camera-ai/gioi-thieu-mot-so-thuat-toan-phat-hien-doi-tuong-pho-bien.html> (Truy cập lần cuối 23/03/2025).
- [8] Phát hiện đối tượng với các khả năng nâng cao của YOLOv8. URL: <https://vi.linux-console.net/?p=34049> (Truy cập lần cuối 23/03/2025).
- [9] toanphan, Tìm hiểu mô hình YOLO từ quá khứ tới hiện tại. URL: <https://tuhoclaptrinhsite.wordpress.com/2023/04/30/tim-hieu-mo-hinh-yolo-tu-qua-khu-toi-hien-tai/> (Truy cập lần cuối 23/03/2025).
- [10] Figure - available from: Mathematical Problems in Engineering. URL:



- [https://www.researchgate.net/figure/Schematic-diagram-of-separating-hyperplanes-for-support-vector-machines\\_fig1\\_361399391](https://www.researchgate.net/figure/Schematic-diagram-of-separating-hyperplanes-for-support-vector-machines_fig1_361399391) (Truy cập lần cuối 12/04/2025).
- [11] Nguyễn Minh Hải, Convolutional Neural Networks là gì? Khám phá cấu trúc của CNN. URL: <https://vnptai.io/vi/blog/detail/convolutional-neural-networks-la-gi> (Truy cập lần cuối 12/04/2025).
- [12] Sidharth, Convolutional Neural Network (CNN): Architecture Explained Deep Learning. URL: <https://www.quarkml.com/2023/06/introduction-to-convolutional-neural-networks.html> (Truy cập lần cuối 16/04/2025).
- [13] Juan R. Terven, Diana M. Cordova-Esparza, A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. URL: <https://arxiv.org/html/2304.00501v6> (Truy cập lần cuối 16/04/2025).
- [14] Representation of linear regression curve based on Eq. 2. URL: [https://www.researchgate.net/figure/Representation-of-linear-regression-curve-based-on-Eq-2\\_fig40\\_356596599](https://www.researchgate.net/figure/Representation-of-linear-regression-curve-based-on-Eq-2_fig40_356596599) (Truy cập lần cuối 16/04/2025).
- [15] Liang-yc, ECUSTFD-resized. URL: <https://github.com/Liang-yc/ECUSTFD-resized-> (Truy cập lần cuối 17/04/2025).