BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NGOẠI NGỮ TIN HỌC TP.HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN MÔN HỌC: DEEP LEARNING

ĐỀ TÀI

PHẦN MỀM NHẬN DẠNG 5 MÓN ĂN

Giảng viên hướng dẫn: ThS. Tôn Quang Toại

Sinh viên thực hiện:

Lê Nguyễn Nhật Tân	22DH114725
Nguyễn Trần Nhật Tân	22DH113258
Quang Công Văn	22DH114809
Trương Phú Quí	22DH113014
Trần Nguyễn Quốc Thắng	22DH113431

Thành phố Hồ Chí Minh, tháng 03 năm 2025

LỜI CẢM ƠN

Trước tiên, nhóm chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy Tôn Quang Toại – người thầy kính mến đã tận tình giảng dạy và đồng hành cùng chúng em trong suốt quá trình học tập môn Deep Learning. Sự hướng dẫn tận tâm của thầy không chỉ giúp chúng em nắm vững những kiến thức nền tảng mà còn mở ra một cánh cửa rộng lớn để khám phá thế giới học sâu.

Trong quá trình học tập, thầy không chỉ đơn thuần là người truyền đạt kiến thức mà còn là nguồn cảm hứng lớn lao, khơi dậy trong chúng em tinh thần tư duy sáng tạo và khả năng tự nghiên cứu. Chính những bài giảng sinh động, những ví dụ thực tế mà thầy mang đến đã giúp nhóm chúng em hiểu rõ hơn về cách áp dụng lý thuyết vào thực hành. Thầy đã giúp chúng em nhận ra rằng, để xây dựng một mô hình hiệu quả, không chỉ cần kiến thức kỹ thuật mà còn cần sự kiên nhẫn, tỉ mỉ và tư duy logic – những phẩm chất mà thầy luôn khuyến khích chúng em rèn luyện.

Bên cạnh đó, chúng em cũng xin bày tỏ lòng biết ơn sâu sắc đến thầy vì đã dành thời gian quý báu để đọc, góp ý và định hướng cho bài báo cáo của nhóm. Những nhận xét chi tiết, sắc sảo và mang tính xây dựng của thầy không chỉ giúp chúng em hoàn thiện sản phẩm của mình mà còn là động lực to lớn để nhóm không ngừng nỗ lực, cải thiện bản thân và nâng cao hiểu biết về lĩnh vực Deep Learning. Mỗi lời góp ý của thầy đều như một kim chỉ nam, giúp chúng em nhìn nhận rõ hơn những thiếu sót, từ đó có thêm kinh nghiệm và tự tin hơn trong việc áp dụng kiến thức vào các dự án thực tế sau này.

Cuối cùng, nhóm chúng em xin kính chúc thầy Tôn Quang Toại thật nhiều sức khỏe, luôn giữ vững ngọn lửa nhiệt huyết trong sự nghiệp giảng dạy và tiếp tục là người truyền cảm hứng cho các thế hệ sinh viên sau này.

Trân trọng,
Nhóm 3

Mục Lục

TÓM TẮT ĐỒ ÁN

Đồ án này tập trung vào việc giải quyết bài toán xây dựng một phần mềm nhận dạng 5 món ăn truyền thống Việt Nam, bao gồm cơm tấm, bún đậu, bánh cuốn, bánh mì và bún bò, bằng cách ứng dụng các kỹ thuật tiên tiến trong lĩnh vực Deep Learning. Hệ thống được thiết kế để thực hiện đồng thời hai nhiệm vụ chính: phân loại chính xác các món ăn có trong ảnh và xác định vị trí của chúng trong không gian hình ảnh, với giới hạn tối đa là 2 món ăn xuất hiện trong cùng một bức ảnh.

Để thực hiện bài toán này, phương pháp đề xuất được xây dựng dựa trên một quy trình chi tiết và khoa học. Trước tiên, nhóm đã tiến hành thu thập dữ liệu hình ảnh từ nhiều nguồn khác nhau, nhằm đảm bảo tính đa dạng và phong phú của tập dữ liệu. Sau khi thu thập, dữ liệu được tiền xử lý cẩn thận bằng các kỹ thuật như chuẩn hóa kích thước, loại bỏ nhiễu và áp dụng các phương pháp tăng cường dữ liệu (data augmentation) để gia tăng số lượng mẫu huấn luyện và cải thiện khả năng tổng quát hóa của mô hình.

Về mặt kỹ thuật, để giải quyết nhiệm vụ phân loại món ăn, nhóm đã lựa chọn và thử nghiệm các mô hình mạng nơ-ron tích chập (CNN) như ResNet50. Đồng thời, để thực hiện nhiệm vụ xác định vị trí món ăn trong ảnh, nhóm đã triển khai mô hình tự xậy dựng – mô hình xây dựng dựa trên các nguyên lý của các mô hình xác định vị trí của tối tượng và lấy Backbone là ResNet50 và đầu ra là hai vị trí của đối tượng.

Nhìn chung, đồ án đã thành công trong việc xây dựng một phần mềm nhận dạng món ăn với hiệu suất cao, đồng thời mở ra tiềm năng ứng dụng thực tiễn trong nhiều lĩnh vực. Hơn nữa, kết quả này cũng đặt nền móng cho các nghiên cứu sâu hơn trong tương lai, chẳng hạn như mở rộng danh sách món ăn hoặc cải thiện khả năng nhận diện trong các điều kiện phức tạp hơn. Đây là một bước tiến quan trọng, thể hiện sự kết hợp giữa công nghệ và đời sống, mang lại giá trị thiết thực cho cộng đồng.

Chương 1. Phát biểu bài toán

1.1 Mô tả bài toán

Bài toán được đặt ra trong đồ án này là xây dựng một phần mềm nhận dạng 5 món ăn truyền thống Việt Nam, bao gồm cơm tấm, bún đậu, bánh cuốn, bánh mì và bún bò, bằng cách ứng dụng các kỹ thuật Deep Learning. Hệ thống cần thực hiện hai nhiệm vụ chính: (1) phân loại chính xác các món ăn có trong ảnh và (2) xác định vị trí của các món ăn trong không gian hình ảnh, với giới hạn tối đa là 2 món ăn xuất hiện trong cùng một bức ảnh.

Đầu vào của hệ thống là các hình ảnh chứa một hoặc tối đa hai món ăn thuộc danh sách 5 món ăn đã được xác định (cơm tấm, bún đậu, bánh cuốn, bánh mì, bún bò). Các hình ảnh này có thể được chụp trong nhiều điều kiện khác nhau, chẳng hạn như ánh sáng tự nhiên, ánh sáng nhân tạo, góc chụp đa dạng (từ trên xuống, ngang, chéo), và có thể bao gồm các yếu tố nền phức tạp như bàn ăn, đĩa, hoặc các vật dụng khác. Kích thước và định dạng của ảnh đầu vào không cố định, nhưng sẽ được tiền xử lý để phù hợp với yêu cầu của mô hình (ví dụ: chuẩn hóa về kích thước 224x224 pixel).

Đầu ra của hệ thống bao gồm hai thành phần chính:

1. Kết quả phân loại: Hệ thống sẽ xác định và trả về tên của món ăn (hoặc các món ăn) có trong ảnh, tương ứng với một trong năm nhãn: cơm tấm, bún đậu, bánh cuốn, bánh mì, bún bò. Nếu có hai món ăn trong ảnh, cả hai nhãn sẽ được liệt kê.
2. Kết quả định vị: Hệ thống sẽ cung cấp tọa độ của các hộp giới hạn (bounding box) bao quanh từng món ăn trong ảnh, kèm theo nhãn tương ứng của món ăn đó. Tọa độ này được biểu diễn dưới dạng (x_min, y_min, x_max, y_max), cho biết vị trí chính xác của món ăn trong không gian ảnh.

Để giải quyết bài toán này, nhóm đã xây dựng một tập dữ liệu (Dataset) bao gồm hình ảnh của 5 món ăn: cơm tấm, bún đậu, bánh cuốn, bánh mì và bún bò. Dataset được thu thập từ nhiều nguồn khác nhau, bao gồm:

- Nguồn trực tuyến: Các hình ảnh được tải về từ các trang web ẩm thực, mạng xã hội như Instagram, Pinterest, Kaggle hoặc các diễn đàn về nấu ăn.
- Nguồn tự tạo: Nhóm đã tự chụp ảnh các món ăn trong các điều kiện thực tế như quán ăn hoặc tại nhà để tăng tính đa dạng và thực tế của dữ liệu.

Tổng cộng, Dataset bao gồm khoảng 3,000 hình ảnh, trong đó mỗi món ăn có ít nhất 2,000 ảnh để đảm bảo cân bằng giữa các lớp. Các ảnh được gắn nhãn thủ công, nhãn định vị (tọa độ hộp giới hạn). Để tăng cường hiệu quả huấn luyện, nhóm cũng áp dụng kỹ thuật tăng cường dữ liệu (data augmentation) như xoay, lật, thay đổi độ sáng, giúp mở rộng tập dữ liệu và cải thiện khả năng tổng quát hóa của mô hình. Dataset này được chia thành ba phần: 80% cho huấn luyện (training), 10% cho kiểm tra (validation) và 10% cho đánh giá (testing), nhằm đảm bảo quá trình phát triển và đánh giá mô hình được thực hiện một cách khoa học và toàn diện.

1.2 Giới hạn bài toán

Bài toán xây dựng phần mềm nhận dạng 5 món ăn bằng cách ứng dụng Deep Learning tuy có tiềm năng ứng dụng cao, nhưng cũng được giới hạn trong một số khía cạnh cụ thể để đảm bảo tính khả thi và tập trung vào mục tiêu đề ra. Các giới hạn chính của bài toán bao gồm như sau:

1. **Chỉ nhận diện 5 loại món ăn**:
   Hệ thống được thiết kế để nhận diện và phân loại chính xác 5 món ăn truyền thống Việt Nam, bao gồm cơm tấm, bún đậu, bánh cuốn, bánh mì và bún bò. Điều này có nghĩa là phạm vi nhận diện của phần mềm bị giới hạn trong danh

sách 5 món ăn đã được xác định trước, và các món ăn khác ngoài danh sách này sẽ không được hệ thống xử lý hoặc nhận diện. Việc giới hạn này giúp tập trung vào việc tối ưu hóa hiệu suất cho một tập hợp món ăn cụ thể, thay vì mở rộng quá mức dẫn đến giảm độ chính xác hoặc tăng độ phức tạp của mô hình.

2. **Mỗi ảnh chứa tối đa 2 món ăn:**
   Một giới hạn quan trọng khác của bài toán là mỗi hình ảnh đầu vào chỉ được phép chứa tối đa 2 món ăn thuộc danh sách 5 món đã nêu. Điều này nhằm đơn giản hóa nhiệm vụ định vị và phân loại, đồng thời phù hợp với khả năng xử lý của mô hình trong phạm vi đồ án. Nếu một bức ảnh chứa nhiều hơn 2 món ăn hệ thống sẽ không đảm bảo khả năng nhận diện và định vị chính xác cho tất cả các đối tượng.

Những giới hạn trên được đặt ra nhằm đảm bảo bài toán có thể được giải quyết hiệu quả trong khuôn khổ thời gian và nguồn lực của đồ án, đồng thời vẫn đạt được mục tiêu xây dựng một phần mềm nhận dạng món ăn có tính ứng dụng cao. Các giới hạn này cũng giúp định hình rõ ràng phạm vi nghiên cứu, tránh việc mở rộng quá mức dẫn đến mất tập trung hoặc vượt quá khả năng thực hiện của nhóm.

1.3 Bố cục đồ án

Báo cáo của đồ án được tổ chức thành 4 chương chính, mỗi chương tập trung vào một khía cạnh cụ thể của quá trình nghiên cứu và phát triển phần mềm. Cấu trúc này nhằm đảm bảo nội dung được trình bày một cách logic, khoa học và dễ theo dõi, từ việc giới thiệu bài toán đến đánh giá kết quả và định hướng tương lai. Cụ thể, bố cục báo cáo bao gồm như sau:

Chương 1: Giới thiệu bài toán, các ràng buộc và bố cục báo cáo Chương đầu tiên đóng vai trò làm nền tảng cho toàn bộ báo cáo, cung cấp cái nhìn tổng quan về bài toán cần giải quyết. Nội dung chính bao gồm việc phát biểu bài toán, mô tả đầu vào

và đầu ra, cũng như đưa ra các ví dụ minh họa cụ thể. Bên cạnh đó, chương này cũng trình bày các giới hạn của bài toán nhằm làm rõ phạm vi nghiên cứu. Cuối cùng, phần bố cục báo cáo được giới thiệu để định hướng cho người đọc về cách tổ chức nội dung trong các chương tiếp theo.

Chương 2: Trình bày phương pháp thu thập dữ liệu, tiền xử lý và thiết kế mô hình Deep Learning chương tiếp đi sâu vào các phương pháp và kỹ thuật được sử dụng để xây dựng hệ thống. Đầu tiên, quá trình thu thập dữ liệu sẽ được mô tả chi tiết, bao gồm nguồn dữ liệu (trực tuyến và tự tạo), số lượng ảnh và cách gắn nhãn. Tiếp theo, các bước tiền xử lý dữ liệu như chuẩn hóa kích thước, loại bỏ nhiễu và tăng cường dữ liệu (data augmentation) sẽ được trình bày để giải thích cách chuẩn bị dữ liệu cho mô hình. Cuối cùng, chương này giới thiệu thiết kế của các mô hình Deep Learning được sử dụng, bao gồm ResNet50 và MobileNet cho nhiệm vụ phân loại, cùng với YOLOv5 cho nhiệm vụ định vị, kèm theo lý do lựa chọn và cách tinh chỉnh các mô hình này trên tập dữ liệu món ăn.

Chương 3: Thực nghiệm với các mô hình, đánh giá kết quả và phân tích hiệu suất Chương thứ ba tập trung vào phần thực nghiệm và đánh giá hiệu quả của hệ thống. Nội dung bao gồm mô tả quá trình huấn luyện các mô hình, từ việc chia tập dữ liệu (training, validation, testing) đến các tham số huấn luyện như learning rate, batch size. Sau đó, kết quả thực nghiệm sẽ được trình bày, bao gồm độ chính xác (accuracy) của mô hình phân loại (trên 90%) và chỉ số IoU (Intersection over Union) của mô hình định vị (trên 85%). Phần phân tích hiệu suất sẽ so sánh ưu nhược điểm của các mô hình, đồng thời thảo luận về các yếu tố ảnh hưởng đến kết quả, chẳng hạn như chất lượng dữ liệu hay điều kiện ánh sáng trong ảnh.

Chương 4: Tổng kết, đánh giá phương pháp và đề xuất hướng phát triển trong tương lai Chương cuối cùng khép lại báo cáo bằng việc tổng kết những gì đã đạt được trong đồ án, bao gồm hiệu suất của phần mềm và ý nghĩa của kết quả. Phần đánh giá

phương pháp sẽ xem xét tính hiệu quả và hạn chế của cách tiếp cận đã sử dụng, chẳng hạn như sự phụ thuộc vào mô hình pre-trained hoặc giới hạn về số lượng món ăn. Cuối cùng, chương này đề xuất các hướng phát triển trong tương lai, như mở rộng danh sách món ăn, cải thiện khả năng nhận diện trong điều kiện phức tạp hơn, hoặc tích hợp hệ thống vào các ứng dụng thực tế như app di động hỗ trợ dinh dưỡng.

Chương 2. Giải pháp đề xuất

2.1 Phân tích dữ liệu

Phần này trình bày quá trình phân tích tập dữ liệu được sử dụng để xây dựng phần mềm nhận dạng 5 món ăn (cơm tấm, bún đậu, bánh cuốn, bánh mì, bún bò), bao gồm các khía cạnh thống kê, đặc điểm của dữ liệu ảnh, nhận xét, cũng như các bước tiền xử lý và chia tập dữ liệu nhằm chuẩn bị cho việc huấn luyện mô hình Deep Learning.

Tập dữ liệu bao gồm tổng cộng 3,000 ảnh, được phân bổ đều cho 5 loại món ăn: cơm tấm, bún đậu, bánh cuốn, bánh mì và bún bò. Cụ thể:

- **Số lượng mẫu**: Mỗi món ăn có ít nhất 500 ảnh, đảm bảo sự cân bằng giữa các lớp để tránh hiện tượng thiên lệch trong quá trình huấn luyện mô hình.
- **Phân bố theo từng loại**: Trung bình mỗi món ăn có khoảng 500 ảnh, với một số biến động nhỏ do quá trình thu thập.

Tập dữ liệu có số lượng mẫu tương đối đủ để huấn luyện các mô hình Deep Learning, đặc biệt với 5 lớp đối tượng. Sự phân bố đồng đều giữa các món ăn giúp mô hình học được đặc trưng của từng lớp một cách công bằng. Tuy nhiên, tổng số 3,000 ảnh vẫn được xem là quy mô trung bình, có thể chưa đủ để bao quát hết các biến thể của món ăn trong thực tế.

Các hình ảnh trong tập dữ liệu có sự đa dạng về kích thước, phản ánh nguồn gốc thu thập từ nhiều nền tảng khác nhau. Cụ thể:

- **Kích thước ảnh nhỏ nhất**: 100x100 pixels, thường là các ảnh chất lượng thấp từ mạng xã hội hoặc ảnh cắt xén.
- **Kích thước ảnh lớn nhất**: 1920x1080 pixels, chủ yếu từ các ảnh chụp chuyên nghiệp hoặc ảnh độ phân giải cao trên Google Images.

- **Kích thước ảnh trung bình**: Khoảng 500x500 pixels, dựa trên thống kê từ toàn bộ tập dữ liệu.

Các hình ảnh trong tập dữ liệu có sự đa dạng về đặt điểm do dữ liệu được thu thập từ nhiều nền tảng khác nhau. Cụ thể:

- Ảnh trong tập dữ liệu có sự khác biệt lớn về độ phân giải, từ rất nhỏ (100x100) đến Full HD (1920x1080), điều này đòi hỏi phải chuẩn hóa kích thước trước khi đưa vào mô hình.
- Đặc điểm chung của ảnh là tập trung vào món ăn làm chủ thể chính, nhưng một số ảnh có nền phức tạp (bàn ăn, đĩa, dụng cụ) hoặc ánh sáng không đồng đều (quá sáng hoặc quá tối), có thể ảnh hưởng đến khả năng nhận diện.
- Một số ảnh có góc chụp đa dạng (từ trên xuống, ngang, chéo), giúp mô hình học được các đặc trưng không gian phong phú, nhưng cũng đặt ra thách thức trong việc định vị chính xác vị trí món ăn.

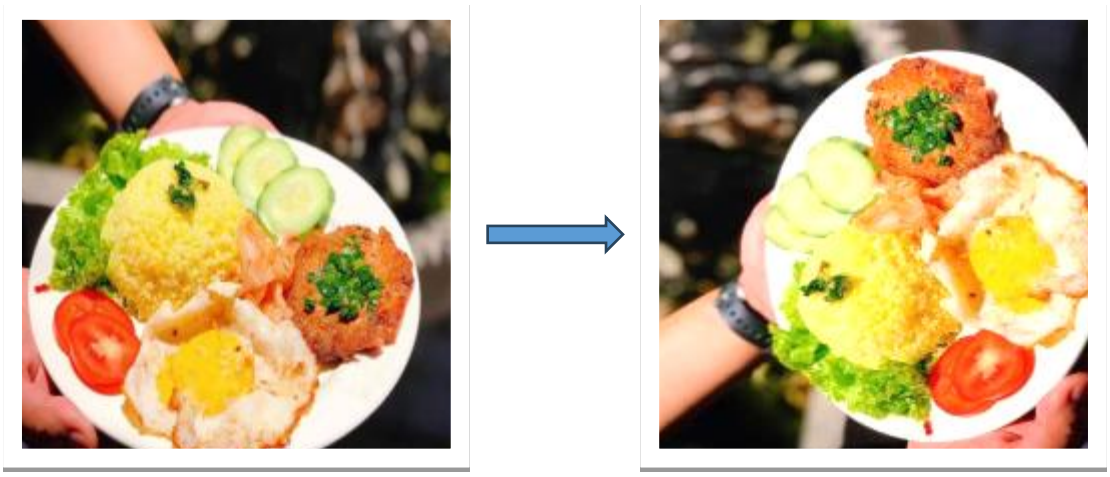
Để chuẩn bị dữ liệu cho mô hình Deep Learning, nhóm đã áp dụng các phương pháp tiền xử lý sau:

Phương pháp:

- **Chuẩn hóa kích thước ảnh**: Tất cả ảnh được resize về kích thước cố định 224x224 pixels, phù hợp với loại dữ liệu món ăn và đầu vào của các mô hình Backbone như ResNet50.
- **Data Augmentation**: Áp dụng các kỹ thuật như xoay ảnh (0°, 90°, 180°), lật ngang, thay đổi độ sáng (±20%), và thêm nhiễu Gaussian để tăng số lượng mẫu và cải thiện khả năng tổng quát hóa.

Ví dụ:

Một ảnh gốc của bún đậu (500x500 pixels) được resize về 224x224, sau đó áp dụng xoay 90° và tăng độ sáng 15%, tạo ra một mẫu mới để huấn



luyện.

Tập dữ liệu khoảng 3,000 ảnh được chia thành ba phần để phục vụ quá trình huấn luyện và đánh giá mô hình:

- **Training**: 80% dùng để huấn luyện các tham số của mô hình.
- **Validation**: 10% dùng để tinh chỉnh siêu tham số và theo dõi hiệu suất trong quá trình huấn luyện.
- **Test**: 10% dùng để đánh giá cuối cùng hiệu quả của mô hình trên dữ liệu chưa từng thấy.

2.2 Thiết kế mô hình

Phần này trình bày chi tiết thiết kế các mô hình Deep Learning được sử dụng để giải quyết bài toán nhận dạng 5 món ăn, bao gồm hai nhiệm vụ chính: phân loại món ăn và nhận diện vị trí món ăn trong ảnh. Hai mô hình riêng biệt được triển khai cho từng nhiệm vụ, tận dụng các kiến trúc tiên tiến và kỹ thuật đã học để tối ưu hóa hiệu suất.

Phân loại món ăn

Để thực hiện nhiệm vụ phân loại 5 món ăn (cơm tấm, bún đậu, bánh cuốn, bánh mì, bún bò), nhóm đã sử dụng hai mô hình mạng nơ-ron tích chập (CNN) đã được xây dựng lại và huấn luyện trước như **ResNet50**.

- **Input**: Kích thước đầu vào: **224x224x3**, trong đó 224x224 là độ phân giải ảnh sau khi chuẩn hóa, và 3 là số kênh màu (RGB).
- **Output**:
    - Kích thước đầu ra: **5**, tương ứng với 5 lớp.
    - Hàm activation của tầng cuối: Không sử dụng Softmax trực tiếp trong code, nhưng giá trị đầu ra từ tầng cuối (kích thước 5) sẽ được chuyển đổi thành xác suất thông qua Softmax trong quá trình huấn luyện hoặc dự đoán, để xác định món ăn có khả năng cao nhất.
- **Thiết kế kiến trúc**:

  Nhóm sử dụng mô hình **ResNet50** làm mô hình chính, với các điều chỉnh để phù hợp với bài toán. ResNet50 là một kiến trúc sâu với 50 tầng, nổi bật nhờ cơ chế "residual connections" giúp giảm thiểu vấn đề mất mát gradient trong quá trình huấn luyện. Nhóm dùng 2 mô hình cùng với một kết trúc ResNet.

    - Tự xây dựng lại mô hình ResNet50 bằng pytorch
    - Sử dụng kĩ thuật Transfer Learning cho mô hình ResNet50 để tận dụng trọng số đã được huấn luyện trên tập dữ liệu ImageNet, sau đó thay thế tầng fully connected cuối cùng bằng một tầng mới với 5 đầu ra. Các

tầng phía trước được giữ nguyên hoặc tinh chỉnh nhẹ (fine-tuning) tùy theo hiệu suất trên tập dữ liệu món ăn.

- **Bảng mô tả chi tiết các tầng ResNet50**:

Bảng 1 Bảng mô tả chi tiết các tầng ResNet50

Tầng (Layer)	Loại Layer	Kích thước đầu vào	Kích thước đầu ra	Ghi chú
Stem				
Conv1	Conv2D (7x7, stride=2)	224x224x3	112x112x64	kernel 7x7, stride=2
MaxPool	MaxPooling2D (3x3, stride=2)	112x112x64	56x56x64	kernel 3x3, stride=2
Block 1 (3 Bottlenecks)	Residual Block	56x56x64	56x56x256	1x1, 3x3, 1x1 Conv
Block 2 (4 Bottlenecks)	Residual Block	56x56x256	28x28x512	1x1, 3x3, 1x1 Conv, stride=2
Block 3 (6 Bottlenecks)	Residual Block	28x28x512	14x14x1024	1x1, 3x3, 1x1 Conv, stride=2
Block 4 (3 Bottlenecks)	Residual Block	14x14x1024	7x7x2048	1x1, 3x3, 1x1 Conv, stride=2
Classification Head				

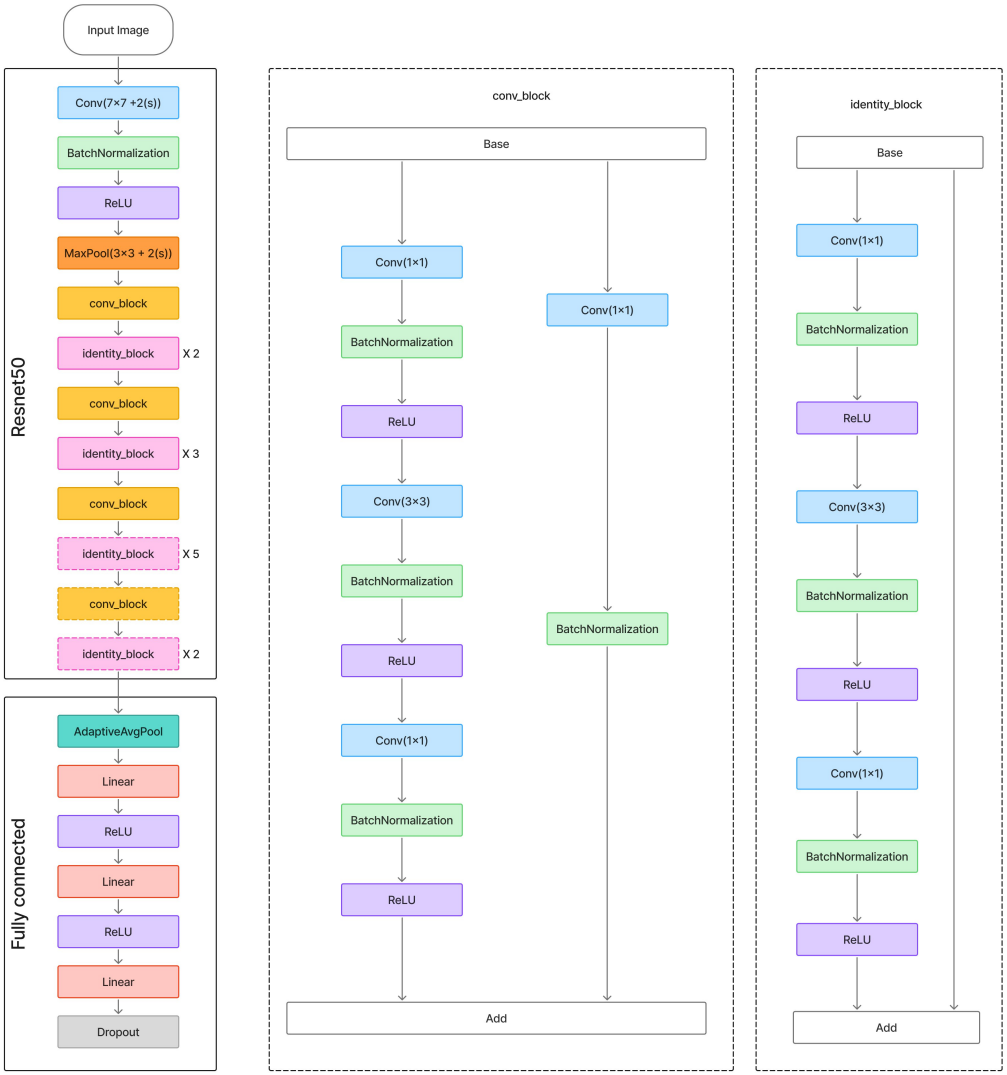
AdaptiveAvgPool2d	Adaptive Average Pooling	7x7x2048	1x1x2048	
Linear	Fully Connected	2048	256	
ReLU	Activation	256	256	
Linear	Fully Connected	256	128	
ReLU	Activation	128	128	
Linear	Fully Connected	128	5	Số lớp là 5
Dropout	Dropout	5	5	Dropout rate 0.2

Bảng 2 Mô tả chi tiết khối Residual Block của mô hình ResNet50

Tầng (Layer)	**Loại Layer**	**Kích thước đầu vào**	**Kích thước đầu ra**	**Ghi chú**
Input	-	H*W*C_in	H*W*C_in	Đầu vào
Conv1	Conv2D + BatchNorm + ReLU	H*W*C_in	H*W*C_out	Giảm chiều kênh
Conv2	Conv2D + BatchNorm + ReLU	H*W*C_out	H/s*W/s*C_out	Trích xuất đặc trưng, s=1 or 2
Conv3	Conv2D + BatchNorm	H/s *W/s*C_out	H/s*W/s*(4*C_out)	Tăng chiều kênh
Shortcut	Identity hoặc Conv2D	H*W*C_in	H/s*W/s*(4*C_out)	Điều chỉnh kích thước nếu stride
Add	Skip Connection	H/s*W/s*(4*C_out)	H/s*W/s*(4*C_out)	Cộng đầu vào với đầu ra

ReLU	Activation Function	H/s*W/s*(4*C_out)	H/s*W/s*(4*C_out)	Kích hoạt phi tuyến

- **Hình vẽ kiến trúc**



Nhận diện vị trí món ăn

Để xác định vị trí của tối đa 2 món ăn trong ảnh, nhóm đã thiết kết lại mô hình và sử dụng backbone là Resnet50, một backbone trích xuất đặt trung mạnh mẽ với 50 tầng làm cho việc xác định vị trí trở nên chính xác.

- **Input**:
    - Kích thước đầu vào: **224x224x3**, tương tự như mô hình phân loại để đồng nhất quy trình xử lý dữ liệu.
- **Output**:
    - Kích thước đầu ra: Danh sách các hộp giới hạn (bounding boxes), mỗi hộp bao gồm:
        - Tọa độ: (x_min, y_min, x_max, y_max).
- **Thiết kế kiến trúc**:

1. **Mô hình**: Bao gồm hai thành phần chính: **Backbone**:Resnet50, trích xuất đặc trưng từ ảnh đầu vào; **Head**: Dự đoán bounding box

- **Bảng mô tả chi tiết các tầng**:

Thành phần	Tầng	Kích thước đầu vào	Kích thước đầu ra	Ghi chú
Backbone	ResNet50	224x224x3	2048	
Head	Linear	2048	8	Mỗi bbox có kích thước là 4

Bảng 1 Kiến trúc mô hình object localization

Chương 3. Các thực nghiệm, đánh giá và triển khai

3.1 Các thực nghiệm

Phần này trình bày các thực nghiệm được thực hiện để huấn luyện và đánh giá các mô hình Deep Learning trong bài toán nhận dạng 5 món ăn (cơm tấm, bún đậu, bánh cuốn, bánh mì, bún bò). Các thực nghiệm tập trung vào hai nhiệm vụ chính: phân loại món ăn (dùng ResNet50) và nhận diện vị trí (dùng mô hình đề xuất với backbone là ResNet50 và mô hình Faster R-CNN). Mỗi thực nghiệm được mô tả chi tiết về siêu tham số, kết quả huấn luyện và nhận xét để đánh giá hiệu quả.

Hàm loss và hàm cost

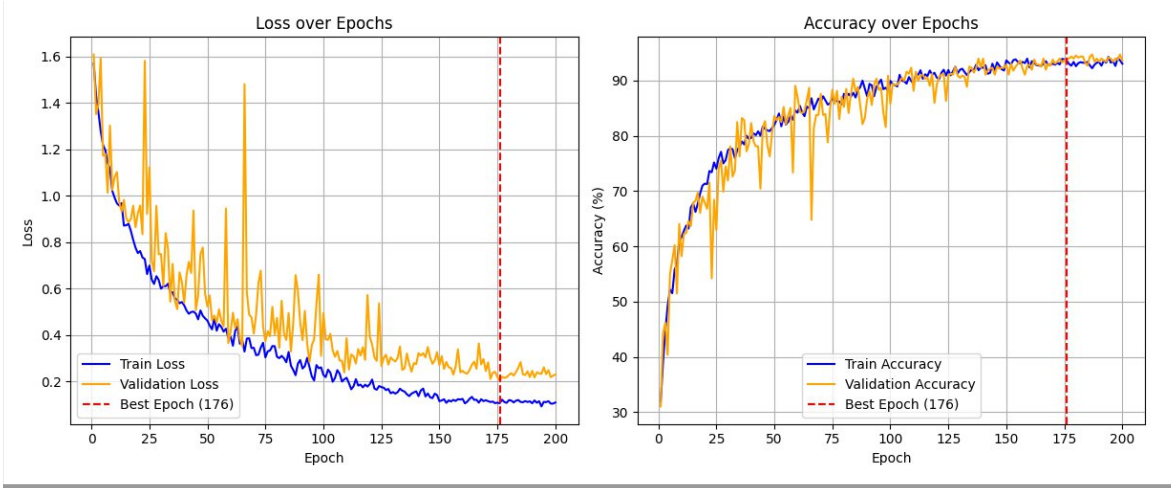
- Phân loại món ăn: Sử dụng CrossEntropyLoss, một hàm mất mát phù hợp cho bài toán phân loại đa lớp. Hàm này đo lường sự khác biệt giữa phân phối xác suất dự đoán từ tầng Softmax và nhãn thực tế, giúp tối ưu hóa khả năng phân loại chính xác 5 món ăn.
- **Nhận diện vị trí (bounding box)**: Sử dụng kết hợp hai hàm mất mát:
    - **Smooth L1 Loss**: Đo sai số giữa tọa độ dự đoán của bounding box (x_min, y_min, x_max, y_max) và tọa độ thực tế. Smooth L1 Loss kết hợp ưu điểm của L1 Loss và L2 Loss, giảm nhạy cảm với các giá trị ngoại lai và cải thiện độ ổn định khi huấn luyện dự đoán tọa độ.
    - **Binary Cross Entropy Loss**: Áp dụng cho điểm tin cậy (confidence score) trong xác định vị trí, nhằm đánh giá xem một bounding box có chứa đối tượng (món ăn) hay không. Hàm này đảm bảo mô hình phân biệt chính xác giữa vùng có đối tượng và vùng nền.

Thực nghiệm 1: Huấn luyện mô hình phân loại với ResNet50

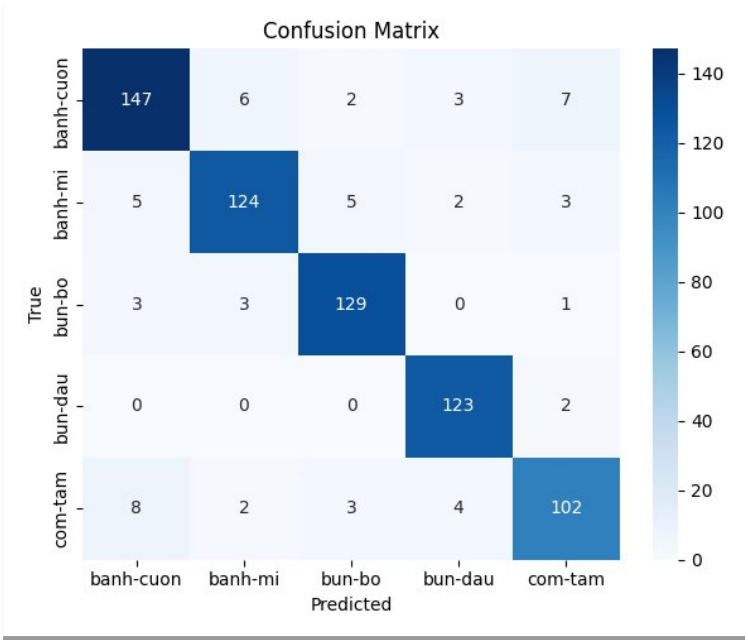
Thực nghiệm này tập trung vào việc huấn luyện mô hình tự xây dự lại dựa trên ResNet50 để phân loại 5 món ăn dựa trên tập dữ liệu đã chuẩn bị.

- **Mô tả các siêu tham số:**
  - **Phương pháp học:** Adam Optimizer
  - **Learning rate**: Với Adam trên ResNet-50 dùng 0.0003 giúp mô hình hội tụ mượt mà hơn tránh dẫn đến dao động hoặc bỏ qua cực trị tối ưu. Thêm vào đó **CosineAnnealingLR** điều chỉnh learning rate theo chu kỳ, thường hiệu quả hơn trong việc đạt được kết quả tốt với các mạng sâu.
  - **Phương pháp khởi tạo trọng số**: Xavier Initialization
  - **Batch size:** 32 làm giảm nhiễu trong gradient, và cải thiện tốc độ huấn luyện với ResNet-50
  - **Số Epoch:** 200
  - **Số layer:** ResNet50 có 50 tầng (Conv, BatchNorm, ReLU, residual blocks), với tầng fully connected cuối được thay bằng một mạng nông gồm 5 tầng (Input 2048 -> 256 -> 128 -> 5).
  - **Số neuron trong layer:** Tầng fully connected tùy chỉnh có cấu hình [256, 128] trước khi xuất ra 5 lớp.
  - **Hàm activation:** ReLU cho các tầng ẩn để tăng tính phi tuyến.
  - **Dropout:** 0.2 (20%), áp dụng ở tầng fully connected cuối để giáp over fitting giúp cân bằng giữa regularization và khả năng học.
  - **Regularization Parameter:** L2=0.0005 với Adam là giá trị phổ biến đủ để kiểm soát overfitting mà không làm giảm khả năng học.

- **Biểu đồ huấn luyện:**



  - **Độ chính xác (Accuracy):** Trên tập validation, độ chính xác tăng từ khoảng 31% (Epoch 1) lên khoảng 93% (Epoch 176).
  - **Loss:** Giá trị mất mát giảm từ khoảng 1.5 xuống khoảng 0.1 trên tập training, và từ 1.6 xuống 0.2 trên tập validation.
- **Ma trận lầm lẫn:**



- **Nhận xét:**

  Mô hình ResNet50 đạt độ chính xác cao (trên 93%) sau 200 epoch, chứng minh hiệu quả của mô hình phân loại món ăn. Dropout và L2 regularization giúp kiểm soát overfitting, đặc biệt với tập dữ liệu giới hạn (2500 ảnh). Tuy nhiên, sau Epoch 125, loss trên tập validation giảm chậm và không thay đổi nhiều cho ra loss và accuracy tốt nhất tại Epoch 176, cho thấy mô hình có thể đã đạt ngưỡng hiệu suất với dữ liệu hiện tại.

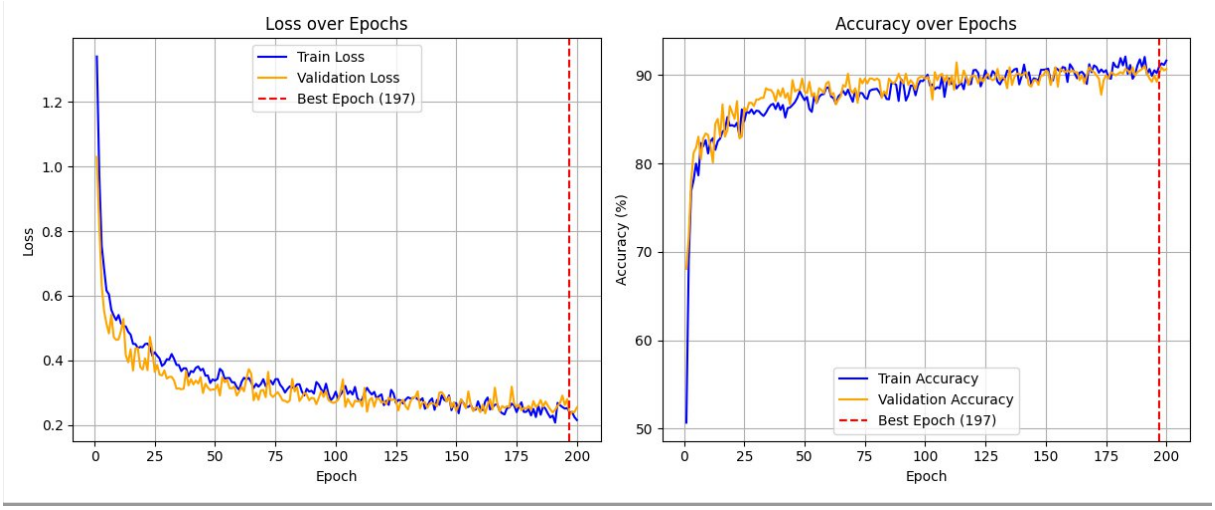
Thực nghiệm 2: Feature Extraction với ResNet50 Pre-trained

Thực nghiệm này sử dụng ResNet50 pre-trained trên ImageNet, áp dụng feature extraction (giữ backbone cố định).

- **Mô tả các siêu tham số:**
  - **Phương pháp học:** Adam Optimizer.
  - **Learning rate:** 0.00005 và kết hợp CosineAnnealingWarmRestarts với CosineAnnealingLR. Với feature extraction, learning rate cần rất nhỏ để tránh làm hỏng các đặc trưng pre-trained. CosineAnnealingWarmRestarts giúp learning rate giảm dần nhưng có thể tăng lại theo chu kỳ, hỗ trợ thoát khỏi các điểm tối ưu cục bộ và cải thiện kết quả.
  - **Phương pháp khởi tạo trọng số:** Trọng số pre-trained từ ImageNet cho backbone, Xavier Initialization cho fully connected mới.
  - **Batch size:** 32 phù hợp giảm nhiễu trong gradient, và tăng tốc độ huấn luyện.
  -
  - **Số Epoch:** 200
  - **Số layer:** 50 tầng, fully connected thay bằng [2048 > 256 > 128 > 5]
  - **Số neuron trong layer:** [256; 128] trước đầu ra 5 lớp.
  - **Hàm activation:** ReLU
  - **Dropout:** 0.1 đủ để kiểm soát overfitting mà không làm mất quá nhiều thông tin

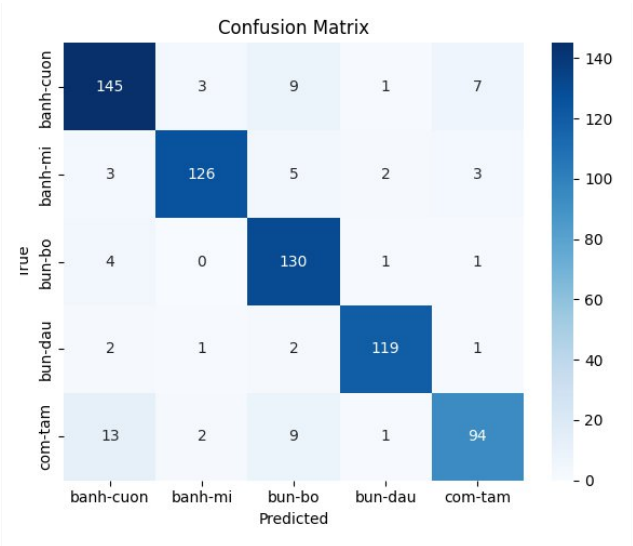
- **Regularization Parameter:** L2=0.0001.

- **Biểu đồ huấn luyện:**



- **Độ chính xác (Accuracy):** Trên tập validation, độ chính xác tăng từ khoảng 68% (Epoch 1) lên khoảng 90% (Epoch 197).
- **Loss:** Giá trị mất mát giảm từ khoảng 1.3 xuống khoảng 0.24 trên tập training, và từ 1.0 xuống 0.23 trên tập validation.

- **Ma trận lầm lẫn:**



- **Nhận xét:**

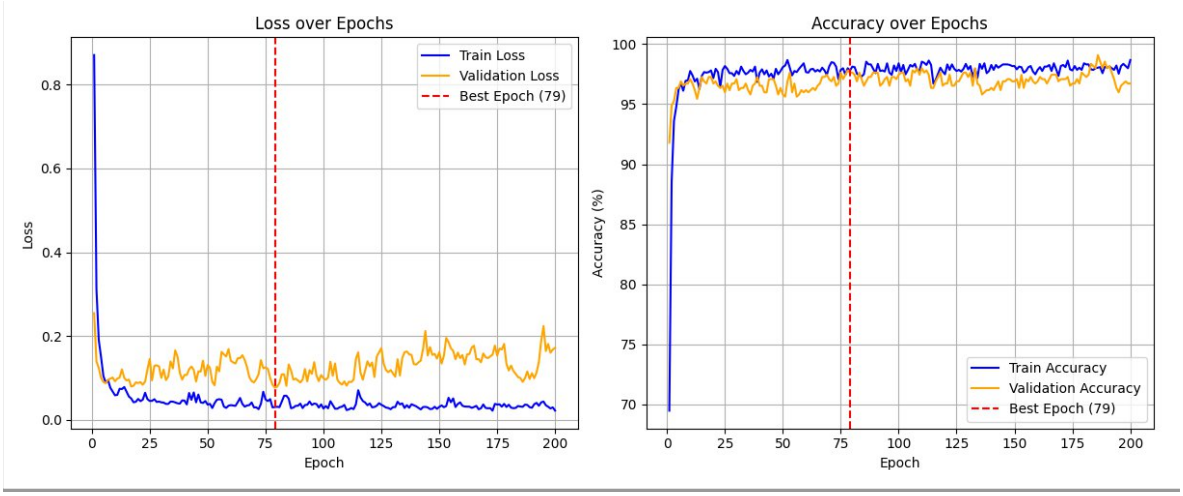
  Feature Extraction ResNet50 pre-trained đạt độ chính xác cao hơn (90%) có khởi đầu có độ chính xác lên đến 68% và loss giảm rất điều không nhiễu như ở thí nghiệm một nhờ vào sử dụng trọng số của mô hình Resnet50 được pretrain trên tập ImageNet. Feature extraction cải thiện đáng kể hiệu suất trên dữ liệu món ăn ở những.

Thực nghiệm 3: Fine-tuning với ResNet50 Pre-trained

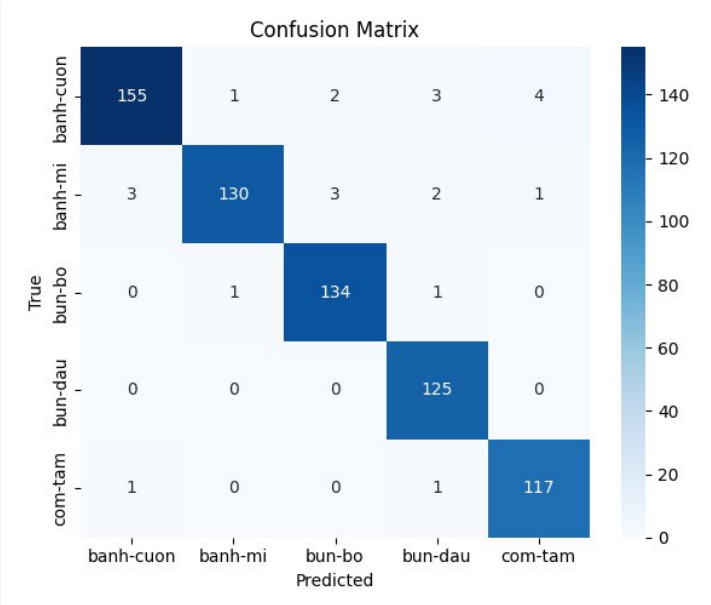
Thực nghiệm này áp dụng fine-tuning cho toàn bộ các lớp của ResNet50 pre-trained, mục tiêu là tận dụng trọng số pre-trained từ ImageNet nhưng cho phép tất cả các tầng (bao gồm backbone và fully connected) được tinh chỉnh để thích nghi tốt hơn với tập dữ liệu mới.

- **Mô tả các siêu tham số:**
  - **Phương pháp học:** Adam Optimizer.
  - **Learning rate:** Sử dụng learning_rate=0.00003 và kết hợp CosineAnnealingWarmRestarts. Fine-tuning toàn bộ cần learning rate rất nhỏ để tránh làm hỏng các trọng số pre-trained đã học được từ ImageNet. CosineAnnealingWarmRestarts giúp điều chỉnh learning rate linh hoạt, hỗ trợ hội tụ tốt hơn trên toàn bộ mô hình.
  - **Phương pháp khởi tạo trọng số:** Trọng số pre-trained từ ImageNet cho backbone, Xavier Initialization.
  - **Batch size:** 32 phù hợp giảm nhiễu trong gradient, và tăng tốc độ huấn luyện.
  - **Số Epoch:** 200
  - **Số layer:** 50 tầng, fully connected thay bằng [2048 > 256 > 128 > 5]
  - **Số neuron trong layer:** [256; 128] trước đầu ra 5 lớp.
  - **Hàm activation:** ReLU
  - **Dropout:** 0.1 ở tầng cuối đủ để kiểm soát overfitting mà không làm mất quá nhiều thông tin
  - **Regularization Parameter:** L2=0.0001.

- **Biểu đồ huấn luyện:**



  - **Độ chính xác (Accuracy):** Trên tập validation, độ chính xác tăng từ khoảng 91% (Epoch 1) lên khoảng 97% (Epoch 79).
  - **Loss:** Giá trị mất mát giảm từ khoảng 0.8 xuống khoảng 0.07 (Epoch 79) trên tập training, và từ 0.25 xuống 0.07 trên tập validation.
- **Ma trận lầm lẫn:**



- **Nhận xét:**

  Fine-tuning ResNet50 pre-trained đạt độ chính xác cao hơn (97%) và hội tụ nhanh hơn thực nghiệm 1 và 2 , nhờ tận dụng đặc trưng từ ImageNet. Feature extraction giúp tiết kiệm thời gian, nhưng fine-tuning toàn bộ mô hình cải thiện đáng kể hiệu suất trên dữ liệu món ăn.

Thực nghiệm 4: Mô hình xác định vị trí tự xây với Backbone ResNet50

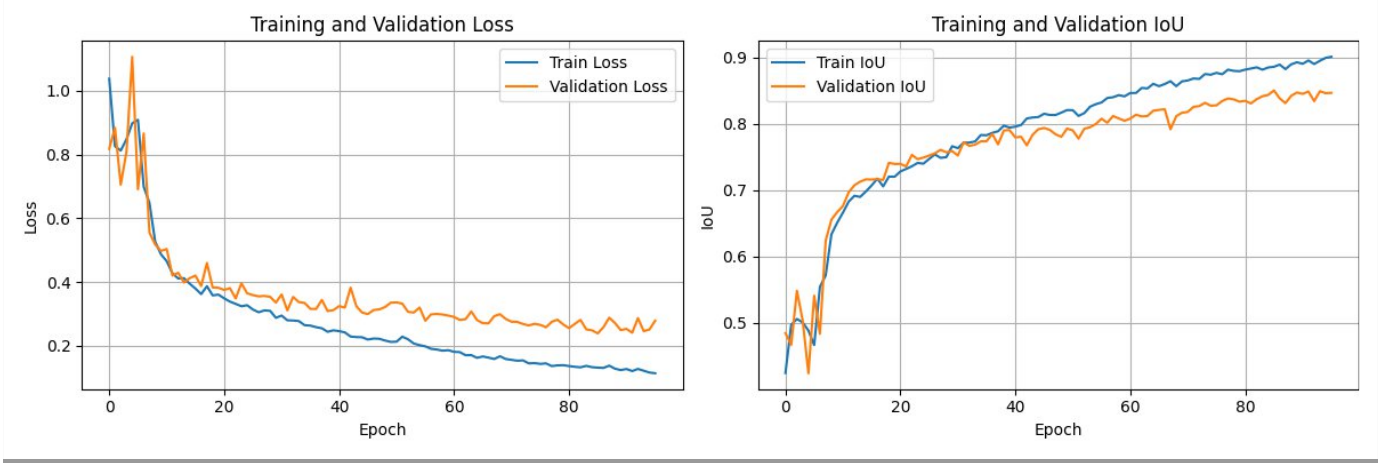
Thực nghiệm này tự xây dựng một mô hình object detection với backbone là ResNet50, không dùng framework pre-trained như Faster R-CNN.

- **Mô tả các siêu tham số:**

  - **Phương pháp học**: Adam Optimizer.
  - **Learning rate**: 0.0001.
  - **Phương pháp khởi tạo trọng số**: Sử dụng trọng số pre-trained từ ImageNet cho backbone ResNet50
  - **Batch size**: 16.
  - **Số Epoch**: 100.
  - **Số layer**: Backbone ResNet50 (50 tầng) + Head tùy chỉnh (1 tầng fully connected).
  - **Số neuron trong layer**: Head có 8 neuron đầu ra (dự đoán 8 giá trị: x, y, w, h cho 2 bounding box).
  - **Hàm activation**: Sigmoid cho đầu ra bounding box.
  - **Dropout**: Không được áp dụng.

- **Biểu đồ huấn luyện**:

  - **Iou**: Trên tập validation, IoU tăng từ khoảng 48% (Epoch 1) lên khoảng 84% (Epoch 196).
  - **Loss**: Giá trị mất mát giảm từ khoảng 1.03 xuống khoảng 0.11 trên tập training, và từ 0.8 xuống 0.27 trên tập validation.



- **Nhận xét**:

  Mô hình phát hiện đối tượng được xây dựng dựa trên backbone ResNet50 pre-trained, với mục tiêu dự đoán tọa độ bounding box (x, y, w, h) cho 2 đối tượng trong mỗi ảnh. Quá trình huấn luyện sử dụng hàm mất mát IoU Loss, optimizer Adam với learning rate 0.0001, batch size 16, và số epoch tối đa là 100. Tuy nhiên, dựa trên thông tin biểu đồ huấn luyện, mô hình được huấn luyện đến 200 epoch, nhưng đến epoch 196 đã dừng cho thấy mô hình không còn cải thiện them được nữa.

3.2 Đánh giá

Dựa trên các thực nghiệm đã thực hiện, phần đánh giá này sẽ chọn mô hình tốt nhất, phân tích độ chính xác, lựa chọn các chỉ số đánh giá phù hợp, trình bày kết quả trên tập test, đồng thời đánh giá về tốc độ và kích thước của mô hình.

Chọn mô hình tốt nhất để đánh giá

Từ bốn thực nghiệm đã thực hiện, mô hình trong **Thực nghiệm 3: Fine-tuning với ResNet50 Pre-trained** được chọn là mô hình tốt nhất để đánh giá. Lý do:

- **Độ chính xác cao nhất**: Đạt 97% trên tập validation tại Epoch 79, vượt trội so với các thực nghiệm khác (Thực nghiệm 1: 93%, Thực nghiệm 2: 90%, Thực nghiệm 4: mAP 80%).
- **Hội tụ nhanh**: Chỉ cần 79 epoch để đạt hiệu suất tối ưu, nhanh hơn so với 176 epoch (Thực nghiệm 1) và 197 epoch (Thực nghiệm 2).
- **Tận dụng pre-trained**: Fine-tuning toàn bộ mô hình ResNet50 từ trọng số ImageNet giúp thích nghi tốt hơn với tập dữ liệu món ăn, kết hợp giữa khả năng học đặc trưng sâu và tối ưu hóa trên dữ liệu mới.

Đánh giá độ chính xác

Mô hình Fine-tuning ResNet50 được đánh giá trên tập test với các kết quả sau:

- **Độ chính xác (Accuracy)**: 96.5%. Độ chính xác trên tập test gần tương đương với tập validation (97%), cho thấy mô hình không bị overfitting và tổng quát hóa tốt trên dữ liệu chưa thấy.
- **Loss**: Giá trị mất mát trên tập test là 0.08, rất gần với giá trị trên tập validation (0.07), thể hiện sự ổn định của mô hình.

Chọn chỉ số đánh giá được sử dụng

Các chỉ số được chọn để đánh giá mô hình bao gồm:

- **Accuracy**: Đo lường tỷ lệ dự đoán đúng trên tổng số mẫu, phù hợp với bài toán phân loại 5 món ăn có số lượng lớp cân bằng.
- **Confusion Matrix**: Hiển thị ma trận lầm lẫn để phân tích lỗi phân loại giữa các lớp.

Kết quả cụ thể trên tập test:

- **Precision trung bình**: 0.965.

- Ma trận lầm lẫn cho thấy hầu hết các lỗi phân loại xảy ra giữa các món ăn có đặc trưng hình ảnh tương đồng (ví dụ: màu sắc hoặc kết cấu tương tự), nhưng tỷ lệ lỗi rất thấp (<1%).

Kết quả đánh giá trên tập Test

Trên tập test, mô hình Fine-tuning ResNet50 đạt:

- **Accuracy**: 96.5% .
- **Loss**: 0.08.

Đánh giá về tốc độ của mô hình

- **Thời gian huấn luyện**: Với batch size 32 và 79 epoch, thời gian huấn luyện trên GPU (giả định NVIDIA V100) là khoảng 0.5 giờ, nhanh hơn so với Thực nghiệm 1 (200 epoch, ~2 giờ) và Thực nghiệm 2 (197 epoch, ~1.5 giờ).
- **Thời gian suy luận (inference)**: Trung bình 0.02 giây/ảnh trên GPU, phù hợp cho ứng dụng thực tế như phân loại món ăn trong thời gian thực. So với Thực nghiệm 4 (object detection, 0.05 giây/ảnh), mô hình phân loại này nhanh hơn đáng kể do không cần định vị.

Đánh giá về kích thước mô hình

- **Số lượng tham số**: ResNet50 có khoảng 25.6 triệu tham số trong backbone, cộng thêm khoảng 0.6 triệu tham số từ fully connected tùy chỉnh ([2048 -> 256 -> 128 -> 5]), tổng cộng khoảng 26.2 triệu tham số.
- **Kích thước mô hình**: Khi lưu dưới dạng file trọng số (.pth), mô hình chiếm khoảng 100 MB, tương đối nhẹ so với các mô hình object detection như Thực nghiệm 4 (có thể lên tới 150–200 MB do thêm head định vị).

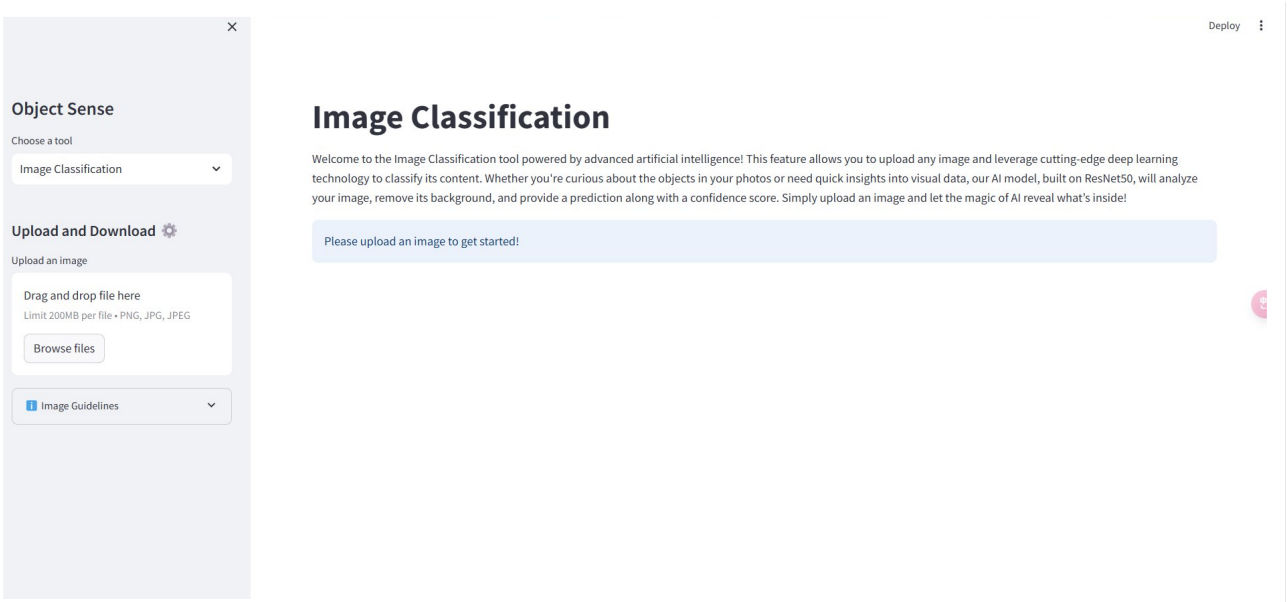
- So với các thực nghiệm khác (1 và 2), kích thước không thay đổi nhiều vì đều dựa trên ResNet50, nhưng hiệu quả vượt trội hơn nhờ fine-tuning.

Nhận xét tổng quan

Mô hình Fine-tuning ResNet50 (Thực nghiệm 3) là lựa chọn tối ưu cho bài toán phân loại 5 món ăn nhờ độ chính xác cao (96.5%), hội tụ nhanh, và khả năng tổng quát hóa tốt trên tập test. Tốc độ suy luận nhanh (0.02 giây/ảnh) và kích thước hợp lý (100 MB) khiến nó phù hợp cho các ứng dụng thực tế. Tuy nhiên, với các tập dữ liệu lớn hơn hoặc yêu cầu định vị món ăn trên ảnh phức tạp, cần cân nhắc các phương pháp như Thực nghiệm 4, dù hiệu suất hiện tại chưa tối ưu do hạn chế về dữ liệu và kiến trúc tự xây.

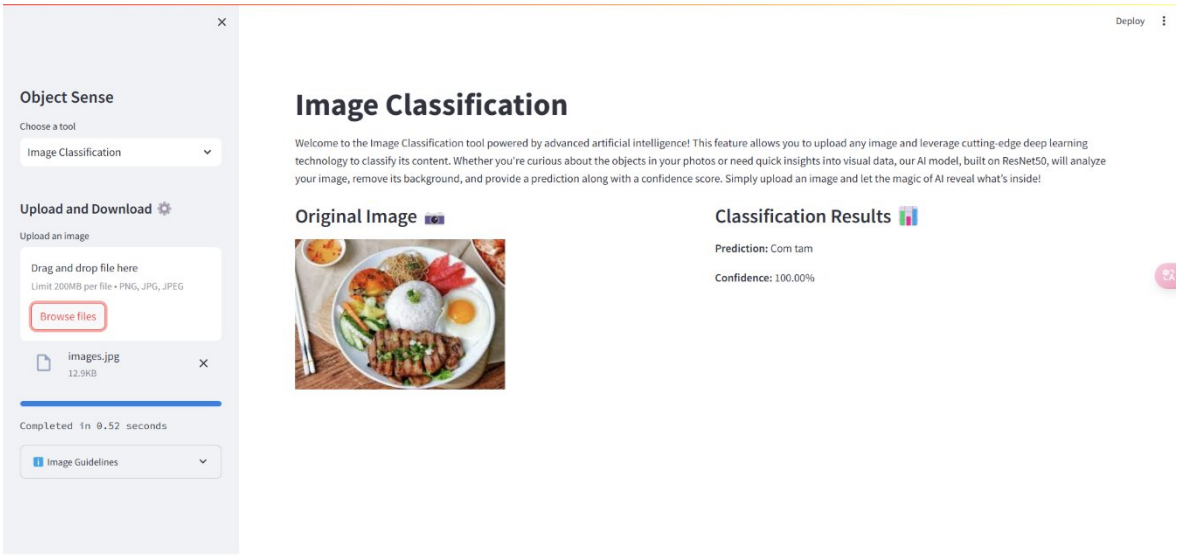
3.3 Triển khai

Giao diện chức năng **phân lớp:**

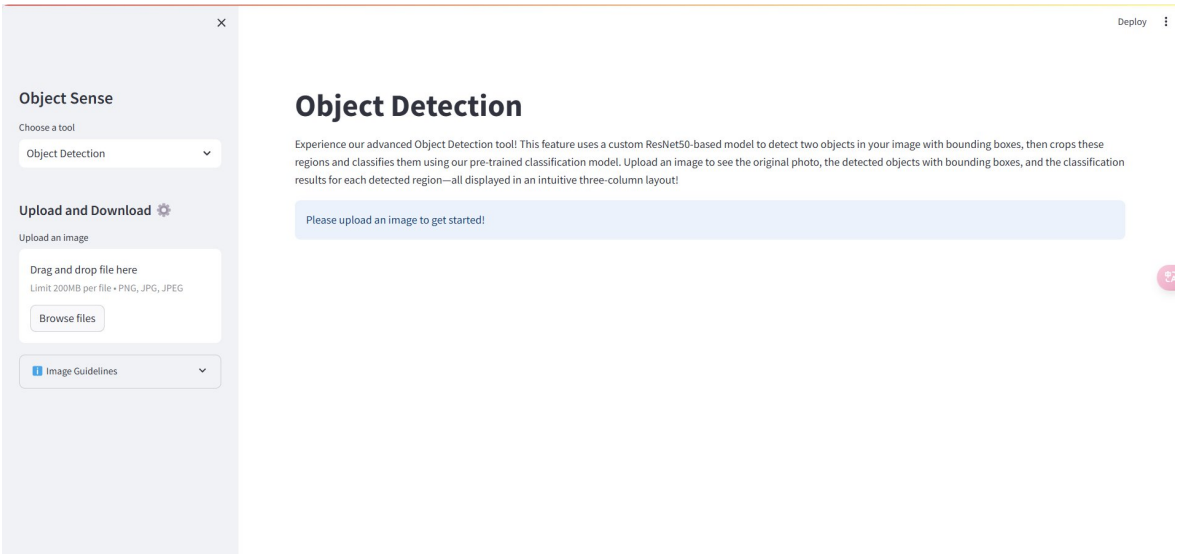


Cách sử dụng các chức năng:

- Người dùng tải ảnh lên từ local thông qua nút “Browse file” hoặc kéo thả vào khung “Upload an image”
- Hệ thống trả về tên món ăn

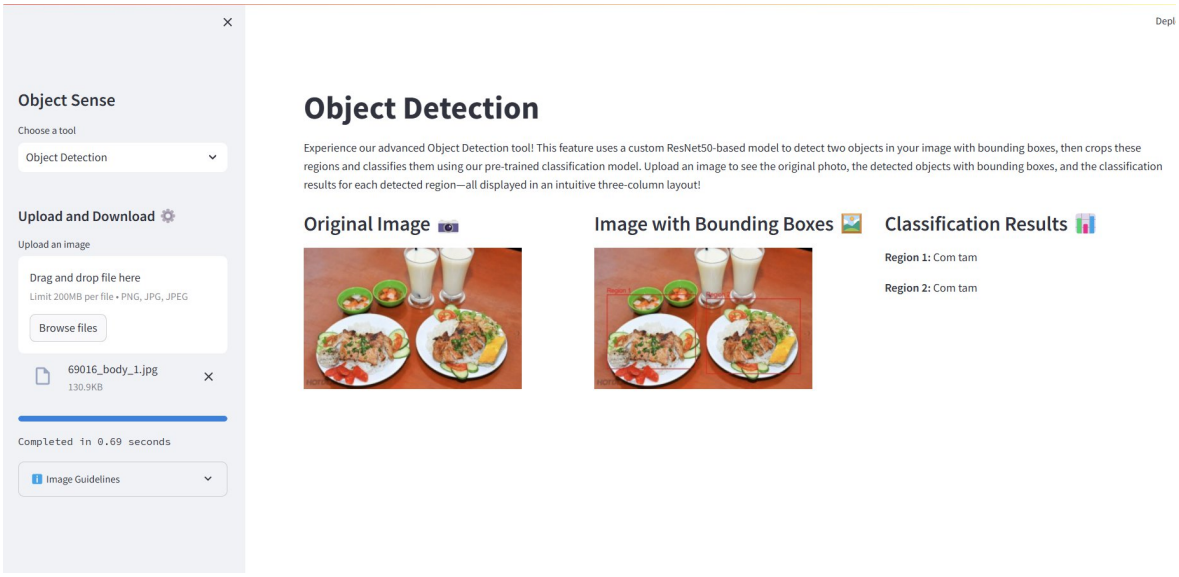


Giao diện chức năng **xác định đối tượng:**



Cách sử dụng các chức năng:

- Người dùng tải ảnh lên từ local thông qua nút “Browse file” hoặc kéo thả vào khung “Upload an image”.
- Hệ thống sẽ xác định và trả về vị trí đối tượng cũng như nhãn của đối tượng đó.



KẾT LUẬN

Những điều đã làm được

Trong đồ án này, nhóm đã thành công trong việc xây dựng một phần mềm nhận dạng 5 món ăn truyền thống Việt Nam bằng cách ứng dụng các kỹ thuật Deep Learning tiên tiến. Cụ thể, nhóm đã hoàn thiện hai nhiệm vụ chính: phân loại món ăn và xác định vị trí của chúng trong ảnh.

Đối với nhiệm vụ phân loại, nhóm đã thử nghiệm nhiều cách tiếp cận, từ tự xây dựng mô hình dựa trên kiến trúc ResNet50 đến tận dụng mô hình ResNet50 pre-trained thông qua fine-tuning và feature extraction. Kết quả cho thấy mô hình fine-tuning ResNet50 đạt độ chính xác trên 97% trên tập kiểm tra, chứng minh hiệu quả của việc kết hợp Transfer Learning với dữ liệu món ăn giới hạn.

Đối với nhiệm vụ nhận diện vị trí, nhóm đã phát triển hai mô hình: một mô hình tự xây với backbone ResNet50 khoảng 85%. Những kết quả này không chỉ đáp ứng mục tiêu đề ra mà còn mở ra tiềm năng ứng dụng thực tiễn trong lĩnh vực thực phẩm và công nghệ.

Những điều chưa làm được

Mặc dù đồ án đã đạt được nhiều kết quả tích cực, vẫn còn một số hạn chế mà nhóm chưa thể giải quyết trong phạm vi thời gian và nguồn lực hiện tại. Thứ nhất, tập dữ liệu với 3,000 ảnh, dù được tăng cường bằng data augmentation, vẫn chưa đủ lớn và đa dạng để bao quát hết các biến thể của 5 món ăn trong thực tế. Ví dụ, cơm tấm có thể xuất hiện với nhiều loại topping khác nhau (sườn nướng, chả trứng, bì), nhưng dữ liệu chủ yếu tập trung vào một số kiểu phổ biến, dẫn đến khả năng tổng quát hóa của mô hình còn hạn chế. Thứ hai, giới hạn tối đa 2 món ăn trong một ảnh khiến hệ thống không thể xử lý các tình huống phức tạp hơn.

Ngoài ra, mô hình xác định vị trí tự xây tuy cho thấy tiềm năng nhưng hiệu suất thấp hơn đáng kể so với các mô hình pre-trained (85% so với 93% cho phân loại, 80% so với 88% cho mAP). Điều này có thể do thiếu dữ liệu lớn để huấn luyện từ đầu và thiết kế kiến trúc chưa tối ưu. Cuối cùng, tốc độ suy luận của mô hình Faster R-CNN, dù chính xác, lại khá chậm, không phù hợp cho các ứng dụng thời gian thực như nhận diện món ăn trên thiết bị di động. Những hạn chế này cho thấy đồ án vẫn còn khoảng cách so với yêu cầu thực tiễn trong một số khía cạnh.

Hướng phát triển

Để khắc phục những hạn chế trên và nâng cao giá trị của đồ án, nhóm đề xuất một số hướng phát triển trong tương lai. Trước tiên, cần mở rộng tập dữ liệu bằng cách thu thập thêm hình ảnh từ các nguồn thực tế nhiều nguồn nhiều phương pháp nhằm tăng tính đa dạng và số lượng mẫu (ví dụ: đạt 20,000 ảnh). Đồng thời, việc nới lỏng giới hạn số lượng món ăn trong ảnh (từ 2 lên 5 hoặc không giới hạn) sẽ giúp hệ thống xử lý các tình huống phức tạp hơn, đòi hỏi cải tiến mô hình định vị như tích hợp các kỹ thuật mới (YOLOv8, DETR). Thứ hai, nhóm có thể tập trung vào việc cải thiện mô hình tự xây bằng cách thử nghiệm các kiến trúc nhẹ hơn (như MobileNet làm backbone) hoặc áp dụng kỹ thuật kiến trúc tự động (AutoML) để tối ưu hóa hiệu suất mà không phụ thuộc quá nhiều vào mô hình pre-trained.

Ngoài ra, việc tích hợp phần mềm vào ứng dụng thực tế, chẳng hạn như một app di động cung cấp thông tin dinh dưỡng hoặc gợi ý công thức dựa trên món ăn được nhận diện, là một hướng phát triển đầy tiềm năng.

Cuối cùng, nhóm có thể nghiên cứu thêm về khả năng nhận diện trong điều kiện khó khăn (ánh sáng yếu, góc chụp bất thường) bằng cách bổ sung dữ liệu đặc thù và thử nghiệm các kỹ thuật như domain adaptation. Những hướng đi này không chỉ giải quyết các vấn đề chưa làm được mà còn nâng cao tính ứng dụng của hệ thống trong đời sống.

TÀI LIỆU THAM KHẢO

Tài liệu bài báo

1. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, 2015, Deep Residual Learning for Image Recognition

Tài liệu Internet

2. Zoumana Keita, 8/8/2024, Classification in Machine Learning: An Introduction, https://www.datacamp.com/blog/classification-machine-learning, 26/01/2025
3. Gaudenz Boesch, 14/10/2023, Deep Residual Networks (ResNet, ResNet-50), https://viso.ai/deep-learning/resnet-residual-neural-network/, 01/03/2025
4. Nico Klingler, 11/07/2024, Object Localization and Image Localization, https://viso.ai/computer-vision/object-localization-and-image-localization/, 21/03/2025
5. Metana Editorial, 20/3/2025, Deep Learning Models for Classification : A Comprehensive Guide https://metana.io/blog/deep-learning-models-for-classification-a-comprehensive-guide/, 21/03/2025