ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN - CƠ - TIN HỌC

TIỂU LUẬN CUỐI KỲ

Image Captioning System for Food

Học phần: Data Mining
Ngành: Khoa học dữ liệu

Nguyễn Ngọc Ánh	MSV: 22001235
Nguyễn Thị Thu Phương	MSV: 22001279
Nguyễn Quang Linh	MSV: 22001270
Phạm Tuấn Hoàng	MSV: 22001260
Nguyễn Văn Dũng	MSV: 22001243

Hà Nội - 2025

Mở đầu

Trong bối cảnh kỷ nguyên số hóa đang phát triển mạnh mẽ như hiện nay, trí tuệ nhân tạo (AI) đã mang lại những bước tiến vượt bậc trong khả năng nhận diện và diễn giải hình ảnh một cách tự động. Khả năng này đóng vai trò thiết yếu trong nhiều lĩnh vực khác nhau như thương mại điện tử, y tế, an ninh mạng, và truyền thông đa phương tiện. Đề tài nghiên cứu của nhóm tập trung vào một ứng dụng chuyên sâu: xây dựng hệ thống "Image Captioning for Food" – một giải pháp tự động tạo ra mô tả bằng ngôn ngữ tự nhiên cho hình ảnh món ăn.

Mục tiêu chính của đề tài là phát triển một hệ thống thông minh có thể tạo ra những mô tả ngữ nghĩa vừa chính xác, vừa giàu ý nghĩa cho ảnh món ăn. Hệ thống không chỉ phân biệt được hình ảnh đó có phải là thực phẩm hay không, mà còn có khả năng sinh ra chú thích chi tiết kèm theo các gợi ý về công thức nấu ăn tương ứng. Đây là sự kết hợp hài hòa giữa các kỹ thuật thị giác máy tính và xử lý ngôn ngữ tự nhiên, mang đến tiềm năng ứng dụng rộng rãi trong các nền tảng tìm kiếm hình ảnh, trợ giúp nấu ăn và các dịch vụ truyền thông về ẩm thực.

Cụ thể, quy trình xử lý dữ liệu của hệ thống gồm ba bước chính. Trước tiên, nhóm sử dụng mô hình ResNet50 được tinh chỉnh để phân loại hình ảnh thành hai nhóm: thực phẩm và phi thực phẩm, giúp lọc ra dữ liệu đầu vào phù hợp cho các bước tiếp theo. Sau đó, với các ảnh thuộc nhóm thực phẩm, mô hình BLIP (Bootstrapped Language-Image Pretraining) sẽ đảm nhiệm nhiệm vụ tự động sinh chú thích (caption) sao cho vừa mạch lạc vừa sát với nội dung ảnh. Cuối cùng, để nâng cao trải nghiệm người dùng, hệ thống sẽ áp dụng mô hình Image-to-Recipe nhằm tạo ra công thức nấu ăn tương ứng dựa trên ảnh món ăn đó. Đặc biệt, nhóm tập trung tinh chỉnh mô hình với dữ liệu ảnh món ăn Việt Nam nhằm tăng khả năng nhận diện và mô tả chính xác những đặc trưng văn hóa ẩm thực bản địa. Bằng cách kết hợp đồng bộ nhiều mô hình học sâu hiện đại trong một pipeline khép kín, đề tài hướng tới mục tiêu xây dựng một hệ thống toàn diện có thể hiểu, phân loại, mô tả và đề xuất công thức cho ảnh món ăn một cách hiệu quả.

Để đảm bảo chất lượng và tính khách quan trong đánh giá, nhóm áp dụng các tiêu chí định lượng phổ biến trong lĩnh vực xử lý ngôn ngữ tự nhiên như BLEU, METEOR, và CIDEr để so sánh và đánh giá độ chính xác của các mô tả sinh ra.

Bảng phân công vai trò

Tên thành viên	Vai trò
Nguyễn Ngọc Ánh	Nhóm trưởng
Phạm Tuấn Hoàng	BA&Test
Nguyễn Thị Thu Phương	Thuật toán
Nguyễn Văn Dũng	Coder
Nguyễn Quang Linh	Cơ sở dữ liệu

Lời cam đoan

Chúng em – nhóm thực hiện đề tài này – xin cam đoan rằng toàn bộ nội dung, kết quả nghiên cứu và sản phẩm trình bày trong báo cáo đều là thành quả thuần túy từ quá trình học tập, nghiên cứu và làm việc nghiêm túc của nhóm. Mọi thông tin, dữ liệu và ý tưởng đều được nhóm tự mình thu thập, phân tích và tổng hợp một cách trung thực và minh bạch.

Chúng em cũng cam kết sử dụng các tài liệu tham khảo và nguồn thông tin bên ngoài một cách rõ ràng, có trích dẫn đầy đủ theo đúng quy định và chuẩn mực học thuật hiện hành.

Trong trường hợp có bất kỳ sai sót, thiếu sót hoặc vi phạm nào được phát hiện trong hoặc sau khi bảo vệ báo cáo, nhóm xin hoàn toàn chịu trách nhiệm trước các cơ quan có thẩm quyền và sẽ chấp nhận mọi hình thức xử lý theo quy định của nhà trường.

Chúng em hy vọng công trình nghiên cứu này sẽ góp phần nhỏ bé vào sự phát triển tri thức trong lĩnh vực khoa học, đồng thời là cơ hội để nhóm nâng cao kỹ năng chuyên môn và khả năng nghiên cứu khoa học của mình.

Trân trọng cảm ơn!

Mục lục

Danh sách hình vẽ

Chương 1

Tổng quan dự án

Chương này nhằm cung cấp cái nhìn tổng quan về bối cảnh, mục tiêu và định hướng phát triển của dự án. Trước tiên, chương sẽ giới thiệu khái quát về bài toán Image Captioning — một trong những bài toán kết hợp giữa thị giác máy tính và xử lý ngôn ngữ tự nhiên. Tiếp theo, chương phân tích động lực phát triển hệ thống trong bối cảnh xã hội hiện đại, nơi mà nhu cầu khám phá thông tin ẩm thực từ hình ảnh ngày càng tăng cao. Bên cạnh đó, phần thực trạng nghiên cứu sẽ điểm lại các cột mốc quan trọng trong hành trình phát triển công nghệ liên quan đến việc mô tả và sinh công thức món ăn từ ảnh. Cuối cùng, chương sẽ trình bày những đóng góp và điểm mới mà dự án của nhóm mang lại, từ đó làm nổi bật giá trị thực tiễn và khả năng ứng dụng của hệ thống trong đời sống.

1.1 Giới thiệu về bài toán

Image Captioning là một bài toán kết hợp giữa thị giác máy tính và xử lý ngôn ngữ tự nhiên, nhằm mục tiêu tạo ra mô tả văn bản phù hợp với nội dung của hình ảnh đầu vào. Image Captioning có nhiều ứng dụng thực tế, chẳng hạn như hỗ trợ người khiếm thị hiểu nội dung hình ảnh, cải thiện khả năng tìm kiếm hình ảnh dựa trên mô tả văn bản, hoặc phát triển hệ thống học nấu ăn tự động từ hình ảnh món ăn.

Để giải quyết bài toán Image Captioning, thường sử dụng kiến trúc encoder-decoder, trong đó encoder (như CNN hoặc Vision Transformer) trích xuất đặc trưng từ ảnh, và decoder (như RNN, LSTM hoặc Transformer) tạo ra mô tả văn bản dựa trên các đặc trưng này.

1.2 Bối cảnh và động lực

Sự bùng nổ của công nghệ số và mạng xã hội đã thay đổi cách mọi người tiếp cận ẩm thực, với việc chia sẻ hình ảnh món ăn trên các nền tảng như Instagram, Pinterest, TikTok trở nên phổ biến. Người dùng không chỉ ghi lại khoảnh khắc ẩm thực tại nhà hàng, trong bữa ăn gia đình hay quá trình tự nấu nướng, mà còn mong muốn khám phá thêm thông tin liên quan như tên món ăn, công thức chế biến, hoặc gợi ý các món tương tự. Điều này phản ánh nhu cầu ngày càng lớn đối với các hệ thống thông minh có khả năng nhận diện, mô tả hình ảnh món ăn một cách tự động và cung cấp các thông tin giá trị, từ hướng dẫn nấu ăn đến gợi ý

sáng tạo.

Hiện nay, khối lượng dữ liệu hình ảnh ẩm thực trên mạng xã hội, blog cá nhân và các nền tảng trực tuyến là vô cùng phong phú, nhưng phần lớn chỉ dừng lại ở hình ảnh mà thiếu đi thông tin chi tiết như công thức, nguyên liệu hay cách chế biến. Trong khi đó, người dùng ngày càng kỳ vọng vào các giải pháp công nghệ đáp ứng nhu cầu thực tế, bao gồm:

- *Khám phá công thức từ hình ảnh*: Nhìn thấy một món ăn hấp dẫn qua ảnh và muốn biết cách tái hiện tại nhà.
- *Nhận diện món ăn và nguyên liệu*: Xác định tên món hoặc gợi ý cách chế biến từ các nguyên liệu sẵn có.
- *Hỗ trợ nấu ăn thông minh*: Sử dụng trợ lý ảo hoặc hệ thống AI để hướng dẫn nấu ăn, cá nhân hóa thực đơn hoặc đề xuất món ăn phù hợp với sở thích và thói quen ăn uống.

Những nhu cầu này không chỉ xuất phát từ sự tò mò về ẩm thực mà còn từ mong muốn tiết kiệm thời gian, nâng cao trải nghiệm nấu nướng và khám phá văn hóa ẩm thực đa dạng. Do đó, việc phát triển một hệ thống AI có khả năng phân tích hình ảnh món ăn, trích xuất thông tin liên quan và cung cấp các gợi ý thực tiễn đang trở thành một bài toán cấp thiết, mang lại giá trị lớn cho cả người dùng cá nhân và ngành công nghiệp ẩm thực.

1.3 Thực trạng nghiên cứu

Bài toán sinh mô tả và công thức nấu ăn từ hình ảnh món ăn đã trải qua một hành trình nghiên cứu dài với nhiều cột mốc quan trọng. Quá trình phát triển có thể chia thành năm giai đoạn chính, từ nhận diện món ăn cơ bản, tiến đến trích xuất nguyên liệu, học biểu diễn đa phương thức, sinh công thức đầy đủ, và cuối cùng là cá nhân hóa ứng dụng trong thực tiễn.

Giai đoạn 1: Nhận diện món ăn

- *Thời gian*: 2010–2015
- *Mô tả*: Giai đoạn đầu tập trung vào việc phân loại món ăn từ hình ảnh, xác định tên món ăn hoặc danh mục (ví dụ: pizza, sushi).
- *Công nghệ*: Các phương pháp truyền thống như túi từ đặc trưng (Bag of Visual Words), SIFT, và máy học cổ điển (SVM). Sau đó, các mạng nơ-ron tích chập (CNN) như AlexNet, VGG được áp dụng và cải thiện hiệu quả đáng kể.
- *Tập dữ liệu*: Food-101, UEC-Food100.
- *Kết quả*: Độ chính xác phân loại đạt 80–90% trên các tập dữ liệu chuẩn, nhưng còn hạn chế với các món có hình thức tương tự.
- *Thách thức*: Thiếu dữ liệu đa dạng, khó nhận diện các món ăn ít phổ biến hoặc bị biến tấu trong trình bày.

Giai đoạn 2: Dự đoán nguyên liệu

- *Thời gian*: 2015–2018
- *Mô tả*: Các nghiên cứu dần mở rộng sang việc dự đoán danh sách nguyên liệu từ hình ảnh món ăn, yêu cầu mô hình suy luận ngữ nghĩa sâu hơn về thành phần.
- *Công nghệ*: CNN (ResNet, Inception), kết hợp học biểu diễn (embedding) để ánh xạ hình ảnh sang không gian nguyên liệu.
- *Tập dữ liệu*: Recipe1M, Yummly-28K.
- *Kết quả*: Độ chính xác dự đoán nguyên liệu đạt 60–70%, nhưng gặp khó khăn với nguyên liệu ẩn (như gia vị, sốt) hoặc trộn lẫn.
- *Thách thức*: Hình ảnh món ăn hoàn chỉnh không thể hiện đầy đủ nguyên liệu, đòi hỏi mô hình phải học được ngữ cảnh và mối quan hệ giữa các thành phần.

Giai đoạn 3: Kết nối ảnh món ăn với công thức

- *Thời gian*: 2017–2020
- *Mô tả*: Đây là giai đoạn quan trọng nhằm xây dựng không gian biểu diễn chung cho ảnh món ăn và công thức nấu ăn (gồm nguyên liệu và hướng dẫn).
- *Công nghệ*: Học nhúng xuyên mô thức (cross-modal embeddings) với cấu trúc hai nhánh: image encoder (CNN) và recipe encoder (biGRU hoặc Transformer). Áp dụng các hàm mất mát như triplet loss, cosine similarity loss để tối ưu khoảng cách giữa các cặp tương ứng.
- *Tập dữ liệu*: Recipe1M.
- *Kết quả*: Mô hình như của Salvador et al. (CVPR 2017 - `https://arxiv.org/pdf/1810.06553v2`) đạt hiệu quả cao trong việc truy xuất công thức từ ảnh và ngược lại.
- *Thách thức*: Mô hình vẫn chưa khai thác hết ngữ nghĩa sâu trong hướng dẫn nấu ăn, và còn bị giới hạn khi biểu diễn các công thức phức tạp.

Giai đoạn 4: Sinh công thức nấu ăn

- *Thời gian*: 2018–2021
- *Mô tả*: Giai đoạn này đánh dấu bước tiến mới khi mô hình có thể sinh ra toàn bộ công thức nấu ăn (tên món, nguyên liệu, các bước chế biến) chỉ từ ảnh món ăn.
- *Công nghệ*: Kiến trúc encoder-decoder (Show-Attend-Tell), mô hình Transformer, và các mô hình đa phương thức như ViLBERT, CLIP. Điển hình là mô hình InverseCooking (2019 - `https://arxiv.org/pdf/1812.06164`) sử dụng ResNet50 + Transformer decoder.

- *Tập dữ liệu*: Recipe1M, RecipeNLG.
- *Kết quả*: Các mô hình có thể tạo ra công thức cơ bản, có tính hợp lý cao trong ngữ cảnh, nhưng thường thiếu chi tiết như liều lượng, thời gian nấu, hoặc kỹ thuật nấu ăn.
- *Thách thức*: Công thức sinh ra dễ bị "ảo giác" (hallucination), và chưa đủ chính xác để dùng trực tiếp trong nấu ăn thực tế.

Giai đoạn 5: Cá nhân hóa và tích hợp thực tiễn

- *Thời gian*: 2021–nay
- *Mô tả*: Nghiên cứu hiện đại hướng đến cá nhân hóa công thức theo khẩu vị, dinh dưỡng, văn hóa; đồng thời tích hợp vào các nền tảng thực tế như trợ lý ảo, hệ thống nấu ăn tự động, hoặc ứng dụng di động.
- *Công nghệ*: Mô hình ngôn ngữ lớn (T5, GPT), học không giám sát, prompting (gợi ý LLMs), học từ dữ liệu mạng xã hội, và tích hợp hệ thống nấu ăn thực tế (như recipe-to-code).
- *Tập dữ liệu*: Recipe1M+, các nguồn không cấu trúc như Instagram, Pinterest, các blog ẩm thực.
- *Kết quả*: Công thức sinh ra có tính tự nhiên hơn, sát với nhu cầu người dùng. Một số hệ thống như FIRE (`https://arxiv.org/pdf/2308.14391`) còn có thể chuyển đổi công thức thành mã điều khiển máy móc.
- *Thách thức*: Cá nhân hóa sâu hơn cho từng vùng miền, ngôn ngữ, điều kiện kinh tế; xử lý món ăn "mới lạ" hoặc không phổ biến; đánh giá chất lượng công thức không chỉ dựa vào ngôn ngữ mà cả khả năng thực thi.

1.4 Đóng góp và điểm mới của dự án

Nhóm dự án tập trung phát triển một hệ thống Image Captioning cho lĩnh vực ẩm thực, tích hợp công nghệ xử lý hình ảnh và ngôn ngữ tự nhiên để cung cấp trải nghiệm tương tác thông minh cho người dùng. Hệ thống có khả năng phân loại ảnh đầu vào thành hai nhóm: ảnh món ăn (Food) và ảnh không phải món ăn (Non-Food), nhằm loại bỏ nhiễu và đưa ra hướng xử lý phù hợp. Với các ảnh được xác định là món ăn, mô hình sẽ sinh mô tả ngắn gọn, tự nhiên và chính xác bằng kỹ thuật image captioning, đồng thời vẫn đảm bảo sinh chú thích phù hợp cho ảnh không thuộc nhóm món ăn.

Điểm nổi bật của hệ thống là khả năng cung cấp thông tin gợi gý công thức cho các ảnh món ăn, bao gồm tên món, danh sách nguyên liệu, hướng dẫn chế biến. Toàn bộ kết quả xử lý được trình bày trên giao diện website thân thiện, dễ sử dụng, hướng đến đối tượng người dùng phổ thông. Với thiết kế linh hoạt và khả năng mở rộng, hệ thống có tiềm năng cao trong

việc ứng dụng thực tế như một công cụ hỗ trợ nấu ăn thông minh, nền tảng gợi ý món ăn theo ảnh hoặc môi trường học nấu ăn tích cực thông qua tương tác trực quan.

1.5 Tổng kết chương và định hướng nội dung tiếp theo

Chương này đã trình bày tổng quan về bài toán sinh mô tả ảnh trong bối cảnh ứng dụng vào lĩnh vực ẩm thực, bao gồm bối cảnh phát triển, động lực triển khai, thực trạng nghiên cứu cũng như đóng góp nổi bật của dự án. Những nội dung này nhằm giúp người đọc hiểu rõ hơn về lý do lựa chọn bài toán, tiềm năng ứng dụng, cũng như các hướng nghiên cứu liên quan đã và đang được quan tâm.

Các chương tiếp theo sẽ đi sâu vào từng khía cạnh cụ thể trong quá trình xây dựng hệ thống:

- **Chương 2 – Thiết kế hệ thống:** Trình bày chi tiết kiến trúc tổng thể, thiết kế cơ sở dữ liệu, xây dựng backend bằng FastAPI, frontend với ReactJS, cùng quy trình đóng gói bằng Docker và triển khai hệ thống trên nền tảng AWS.
- **Chương 3 – Triển khai hệ thống:** Bao gồm quy trình thu thập và xử lý dữ liệu, triển khai và tinh chỉnh các mô hình học sâu (ResNet50, BLIP, Image-to-Recipe), cũng như xây dựng website tích hợp các mô hình này.
- **Chương 4 – Đánh giá thực nghiệm kiểm thử:** Phân tích chi tiết quy trình kiểm thử mô hình và hệ thống, từ thiết lập môi trường, xác định mục tiêu cho đến kết quả kiểm thử nhằm đánh giá hiệu năng và độ ổn định của hệ thống.
- **Chương 5 – Kết luận và hướng phát triển:** Tổng kết các kết quả đạt được, các khó khăn trong quá trình thực hiện và đề xuất các hướng mở rộng hệ thống trong tương lai.
- **Phụ lục – A, B, C:** Cung cấp tài liệu hỗ trợ gồm tập dữ liệu, mã nguồn hệ thống, và hướng dẫn sử dụng sản phẩm một cách chi tiết.

Như vậy, phần tiếp theo sẽ đi vào khía cạnh kỹ thuật đầu tiên: thiết kế hệ thống – nhằm hiện thực hóa các mục tiêu đã đề ra trong chương này.

Chương 2

Thiết kế hệ thống

Chương này mô tả cách nhóm phát triển một hệ thống sinh mô tả món ăn từ ảnh, bao gồm giao diện người dùng, xử lý dữ liệu phía sau, và phần triển khai. Thiết kế hệ thống dựa trên ba thành phần chính: frontend, backend tích hợp mô hình AI, và tầng lưu trữ dữ liệu. Kiến trúc này nhằm đảm bảo hệ thống có thể mở rộng, dễ bảo trì và hoạt động hiệu quả. Ngoài ra, chương còn đề cập đến việc lựa chọn mô hình học sâu, công nghệ sử dụng, cũng như cách triển khai hệ thống trên nền tảng đám mây phù hợp với nhu cầu thực tế.

2.1 Thiết kế kiến trúc tổng quan

Hệ thống được xây dựng theo mô hình client-server truyền thống, kết hợp thêm các thành phần xử lý bằng AI. Khi người dùng tải ảnh món ăn, hệ thống sẽ tiếp nhận, phân tích và sinh ra gợi ý công thức nhờ vào mô hình học sâu. Tổng thể hệ thống được chia làm ba lớp: giao diện người dùng, xử lý nghiệp vụ và AI, và lưu trữ dữ liệu.

a. Tầng giao diện người dùng

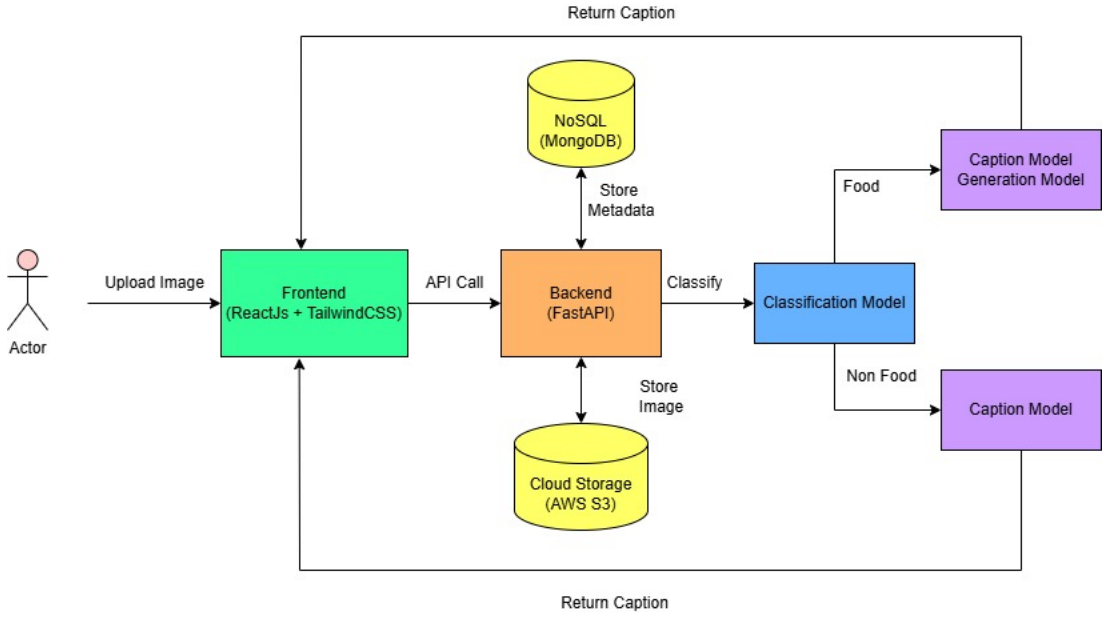
Giao diện web cho phép người dùng tải ảnh món ăn và nhận phản hồi là công thức hoặc mô tả món ăn. Giao tiếp giữa frontend và backend được thực hiện thông qua các API. Giao diện cũng hiển thị danh sách nguyên liệu và các bước thực hiện dưới dạng trực quan.

b. Tầng xử lý nghiệp vụ và AI

Phần backend đảm nhận việc xác thực, điều phối dữ liệu và truyền ảnh đến mô hình AI. Mô hình sẽ phân tích và sinh mô tả ảnh cũng như gợi ý công thức, sau đó gửi kết quả về cho giao diện hiển thị.

c. Tầng lưu trữ dữ liệu

Thông tin người dùng và kết quả mô tả món ăn được lưu trong cơ sở dữ liệu dạng NoSQL. Ảnh do người dùng tải lên sẽ được lưu vào hệ thống lưu trữ đối tượng, giúp dễ dàng quản lý và truy xuất.

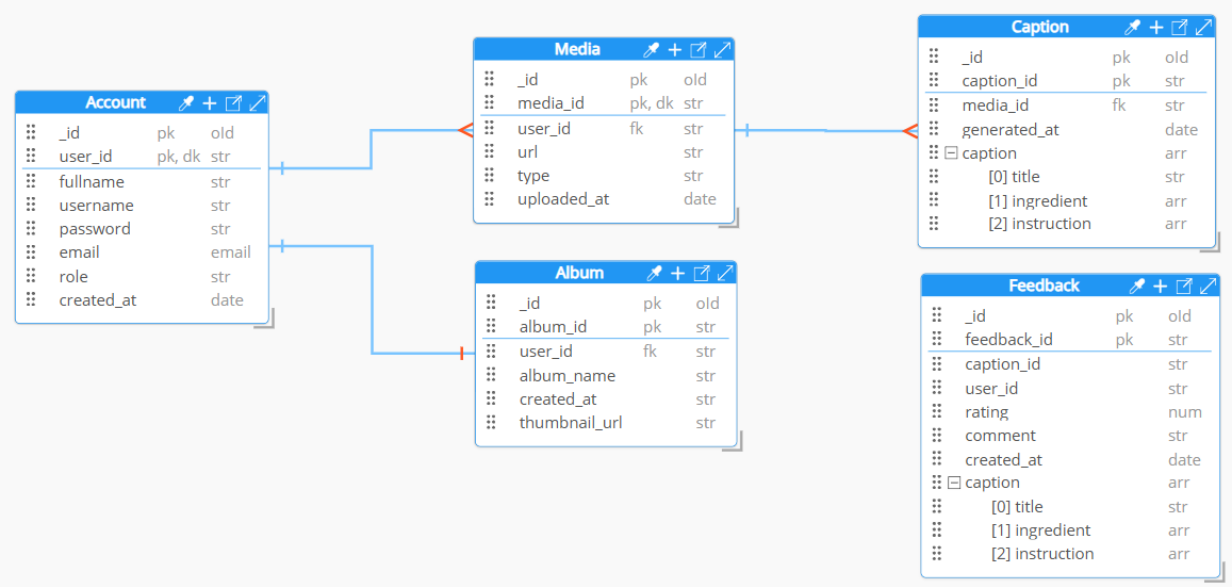


Hình 2.1: Tổng quan kiến trúc hệ thống

2.2 Thiết kế cơ sở dữ liệu

Trong quá trình phát triển hệ thống, nhóm quyết định triển khai một cơ sở dữ liệu có cấu trúc đơn giản nhưng hiệu quả, tập trung vào việc lưu trữ và quản lý các thông tin cốt lõi bao gồm: người dùng, hình ảnh, bộ sưu tập, mô tả (caption) và đánh giá (feedback). Thiết kế này đảm bảo tính dễ hiểu, dễ triển khai, đồng thời vẫn giữ được khả năng mở rộng khi hệ thống phát triển trong tương lai.

Thay vì xây dựng một hệ thống quan hệ phức tạp với nhiều bảng phụ, nhóm lựa chọn mô hình dữ liệu hướng tài liệu (document-based), phù hợp với các ứng dụng web hiện đại, giúp quá trình truy xuất và thao tác dữ liệu diễn ra nhanh chóng, linh hoạt. Hình 2.2 minh họa lược đồ tổng quát của cơ sở dữ liệu được sử dụng.



Hình 2.2: Lược đồ tổng quát cơ sở dữ liệu của hệ thống

Quản lý người dùng (*Users*)

Bộ sưu tập này lưu trữ thông tin về các tài khoản người dùng đã đăng ký trong hệ thống. Thông tin được lưu bao gồm định danh, thông tin xác thực và vai trò của người dùng.

- *user_id*: Mã định danh duy nhất của người dùng trong hệ thống.
- *user_name*: Tên đăng nhập (username), dùng để đăng nhập hệ thống.
- *full_name*: Họ tên đầy đủ của người dùng, được hiển thị trong giao diện.
- *pass_word*: Mật khẩu đã được mã hóa để đảm bảo tính bảo mật.
- *email*: Địa chỉ thư điện tử dùng cho liên hệ và khôi phục tài khoản.
- *role*: Vai trò của người dùng, ví dụ: `admin`, `regular`.
- *created_at*: Thời điểm người dùng tạo tài khoản.

Quản lý hình ảnh (*Media*)

Bộ sưu tập này lưu trữ thông tin về các hình ảnh được tải lên bởi người dùng. Mỗi hình ảnh đều được liên kết với người dùng và bộ sưu tập chứa nó.

- *media_item*: Định danh duy nhất cho từng ảnh.
- *user_id*: Mã người dùng đã tải ảnh.
- *album_id*: Mã bộ sưu tập chứa ảnh.
- *url*: Đường dẫn ảnh được lưu trên hệ thống (hoặc cloud).

- *uploaded_ at*: Thời gian ảnh được tải lên.

Tổ chức bộ sưu tập (*Albums*)

Bảng này cho phép người dùng nhóm các hình ảnh theo từng bộ sưu tập. Một người dùng có thể tạo nhiều bộ sưu tập khác nhau để quản lý ảnh hiệu quả hơn.

- *album_ id*: Định danh duy nhất của bộ sưu tập.
- *user_ id*: Mã người dùng sở hữu bộ sưu tập.
- *album_ name*: Tên bộ sưu tập do người dùng đặt.
- *created_ at*: Thời gian tạo bộ sưu tập.
- *thumbnail_ url*: Đường dẫn ảnh đại diện của bộ sưu tập.

Quản lý mô tả(*Captions*)

Sau khi ảnh được tải lên, hệ thống sẽ sinh tự động mô tả bằng mô hình học sâu. Các mô tả này được lưu lại để phục vụ cho hiển thị và đánh giá.

- *caption_ id*: Định danh duy nhất của mô tả.
- *media_ id*: Liên kết đến ảnh được mô tả.
- *generated_ at*: Thời điểm hệ thống sinh mô tả.
- *caption*: Nội dung mô tả do hệ thống tạo ra.

Quản lý đánh giá của người dùng (*Feedback*)

Hệ thống cho phép người dùng đánh giá chất lượng mô tả được sinh ra. Những phản hồi này giúp cải thiện mô hình sinh mô tả trong các phiên bản tiếp theo.

- *feedback_ id*: Mã định danh của phản hồi.
- *caption_ id*: Mã mô tả được đánh giá.
- *user_ id*: Mã người dùng thực hiện đánh giá.
- *rating*: Điểm số đánh giá (thang điểm từ 1 đến 5).
- *comment*: Nhận xét chi tiết của người dùng.
- *created_ at*: Thời điểm gửi đánh giá.
- *caption*: Nội dung mô tả được người dùng đánh giá.

2.3 Lựa chọn hệ quản trị cơ sở dữ liệu

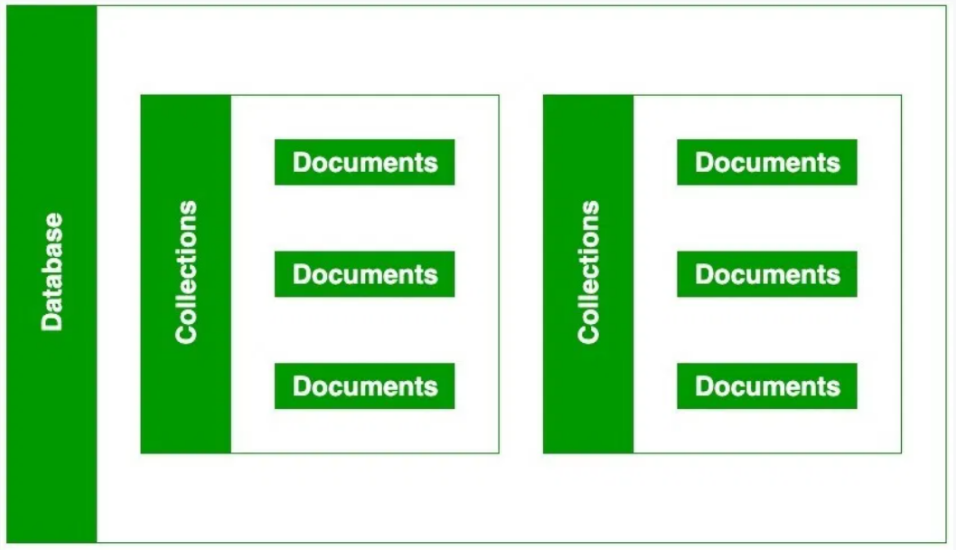
Việc lựa chọn hệ quản trị cơ sở dữ liệu phù hợp đóng vai trò quan trọng trong việc đảm bảo hiệu quả lưu trữ, truy xuất và mở rộng hệ thống. Xét trên đặc điểm của bài toán, nhóm quyết định sử dụng MongoDB – một hệ quản trị cơ sở dữ liệu NoSQL – nhằm đáp ứng tốt yêu cầu linh hoạt về cấu trúc dữ liệu cũng như khả năng mở rộng trong tương lai.

Trong hệ thống này, mỗi mục dữ liệu do người dùng nhập vào gồm ba thành phần chính: `title` (tiêu đề món ăn), `ingredient` (danh sách nguyên liệu), và `instruction` (các bước hướng dẫn chế biến). Cả hai trường `ingredient` và `instruction` đều là các danh sách (arrays) có độ dài thay đổi tùy vào từng món ăn cụ thể, đồng thời có thể phát sinh thêm thuộc tính trong tương lai như khối lượng, đơn vị tính, hoặc thời gian nấu. Do đó, hệ quản trị cơ sở dữ liệu quan hệ truyền thống sẽ khó đáp ứng linh hoạt khi cần mở rộng schema.

Ngoài ra, định hướng mở rộng của hệ thống trong giai đoạn tiếp theo như sinh gợi ý dinh dưỡng, ước lượng calo hoặc truy xuất công thức nấu ăn từ hình ảnh đầu vào, đòi hỏi một cơ sở dữ liệu có khả năng mở rộng linh hoạt, ít ràng buộc về schema, và có thể lưu trữ cả dữ liệu có cấu trúc và bán cấu trúc. Điều này khiến MongoDB trở thành lựa chọn phù hợp và chiến lược cho hệ thống.

a. Tổng quan về MongoDB

MongoDB là một trong những hệ quản trị cơ sở dữ liệu NoSQL phổ biến nhất hiện nay. Thay vì lưu trữ dữ liệu theo mô hình bảng như trong cơ sở dữ liệu quan hệ, MongoDB sử dụng mô hình tài liệu (document model), trong đó mỗi bản ghi được biểu diễn dưới dạng một tài liệu JSON mở rộng (BSON). Mô hình này cho phép dữ liệu có thể linh hoạt về cấu trúc, dễ dàng thay đổi hoặc bổ sung thuộc tính mà không làm ảnh hưởng đến toàn bộ hệ thống.



Hình 2.3: Các thành phần dữ liệu cơ bản trong MongoDB

Trong MongoDB, dữ liệu được tổ chức thành các *collection* (tương đương với bảng trong RDBMS), và mỗi *collection* chứa nhiều *document* có thể có cấu trúc linh hoạt. Mỗi document bao gồm các trường dữ liệu theo dạng cặp khóa–giá trị, trong đó giá trị có thể là chuỗi, số, mảng, đối tượng lồng nhau, v.v. Cách tổ chức này đặc biệt phù hợp với các ứng dụng yêu cầu xử lý dữ liệu phi cấu trúc hoặc có định dạng thay đổi theo thời gian.

b. Ưu điểm nổi bật của MongoDB

- **Tính linh hoạt về schema:** MongoDB cho phép các document trong cùng một collection có cấu trúc khác nhau, rất phù hợp với các bài toán có dữ liệu đầu vào đa dạng và có thể thay đổi trong tương lai.
- **Khả năng mở rộng tốt:** MongoDB hỗ trợ mở rộng theo chiều ngang (horizontal scaling) thông qua cơ chế *sharding*, giúp phân phối dữ liệu trên nhiều máy chủ nhằm đảm bảo hiệu suất và khả năng chịu tải của hệ thống.
- **Hiệu năng cao:** Cơ chế lưu trữ theo document và hỗ trợ chỉ mục mạnh mẽ giúp MongoDB truy vấn dữ liệu nhanh, đặc biệt là khi xử lý các truy vấn động hoặc dữ liệu có cấu trúc phức tạp.
- **Tính khả dụng cao:** MongoDB hỗ trợ tự động sao lưu và phục hồi dữ liệu thông qua *replica set*, giúp đảm bảo tính liên tục và an toàn của dữ liệu trong trường hợp lỗi hệ thống.
- **Hệ sinh thái phong phú và dễ tích hợp:** MongoDB cung cấp nhiều thư viện và công cụ hỗ trợ với các ngôn ngữ lập trình phổ biến, cùng với các công cụ quản lý trực quan như MongoDB Compass.

c. Tích hợp với hệ thống

Trong quá trình xây dựng hệ thống, MongoDB được tích hợp trực tiếp với backend thông qua thư viện `motor` – một driver bất đồng bộ chính thức của MongoDB dành cho Python. Khi kết hợp với FastAPI (một web framework bất đồng bộ), mô hình này cho phép xử lý song song nhiều yêu cầu truy vấn cơ sở dữ liệu, giúp giảm thiểu độ trễ và nâng cao hiệu năng tổng thể. Mỗi thao tác thêm, sửa, truy vấn hay xóa dữ liệu đều được xử lý theo cơ chế bất đồng bộ (*asynchronous*), tối ưu hóa khả năng phục vụ của hệ thống trong môi trường đa người dùng hoặc yêu cầu thời gian thực.

2.4 Thiết kế và phát triển backend với FastAPI

Phần backend của hệ thống được phát triển bằng ngôn ngữ Python, sử dụng FastAPI – một framework hiện đại, nổi bật với khả năng xây dựng API mạnh mẽ, hiệu quả và dễ mở

rộng. FastAPI được chọn không chỉ vì tốc độ xử lý vượt trội mà còn vì sự hỗ trợ tốt cho các ứng dụng cần xử lý song song nhiều yêu cầu.

Một số ưu điểm nổi bật của FastAPI giúp nâng cao hiệu quả phát triển backend bao gồm:

- *Xử lý bất đồng bộ (asynchronous):* Dựa trên tính năng async của Python, FastAPI có thể xử lý hàng ngàn request đồng thời mà không bị nghẽn, giúp hệ thống hoạt động mượt mà hơn, đặc biệt với các tác vụ như tải ảnh hay truy vấn dữ liệu lớn.
- *Tự động xác thực và kiểm tra dữ liệu đầu vào:* Nhờ tích hợp với thư viện Pydantic, các yêu cầu từ phía người dùng được kiểm tra kỹ lưỡng trước khi xử lý. Điều này không chỉ đảm bảo dữ liệu đầu vào hợp lệ mà còn giúp phát hiện lỗi sớm và giảm tải công việc kiểm tra thủ công.
- *Tài liệu API được sinh tự động:* Khi phát triển, FastAPI tự động tạo ra các trang tài liệu tương tác với giao diện thân thiện thông qua Swagger UI và ReDoc. Nhờ đó, việc kiểm thử và trình bày API trở nên trực quan, tiết kiệm thời gian cho cả nhóm phát triển và các bên liên quan.
- *Tuân thủ kiến trúc RESTful:* Các chức năng được phân chia rõ ràng thành các route độc lập, giúp hệ thống dễ đọc, dễ bảo trì và thuận tiện mở rộng trong tương lai.
- *Khả năng tích hợp dễ dàng với các thư viện học máy:* Do phát triển hoàn toàn bằng Python, FastAPI có thể kết hợp chặt chẽ với các thư viện như TensorFlow, PyTorch, giúp việc triển khai các mô hình học sâu trở nên liền mạch và hiệu quả.
- *Cơ chế bảo mật linh hoạt:* Việc sử dụng kết hợp JWT và OAuth2 giúp hệ thống xác thực người dùng một cách an toàn và hiện đại, đảm bảo rằng chỉ những người dùng hợp lệ mới có quyền truy cập vào các chức năng nhạy cảm của hệ thống.

2.5 Thiết kế và phát triển giao diện người dùng với ReactJS

Giao diện người dùng được xây dựng dựa trên thư viện ReactJS – một công cụ mạnh mẽ và phổ biến để phát triển ứng dụng web hiện đại theo kiến trúc component. Với ReactJS, giao diện được thiết kế trực quan, phản hồi nhanh và dễ bảo trì.

Một số lợi thế khi sử dụng ReactJS cho frontend bao gồm:

- *Cập nhật giao diện nhanh nhờ Virtual DOM:* React sử dụng cơ chế Virtual DOM để chỉ cập nhật những phần cần thiết trên giao diện thay vì toàn bộ trang, từ đó giảm thời gian phản hồi và cải thiện trải nghiệm người dùng.
- *Quản lý trạng thái hiệu quả:* Thư viện cung cấp các công cụ như React Hooks và Redux để quản lý trạng thái ứng dụng một cách mạch lạc, đặc biệt hữu ích trong các tương tác phức tạp và nhiều dữ liệu.

- *Tái sử dụng component:* Các phần giao diện được xây dựng dưới dạng component độc lập, có thể tái sử dụng ở nhiều nơi, giúp giảm thiểu việc lặp lại mã và dễ dàng kiểm soát thay đổi.
- *Mở rộng linh hoạt:* React cho phép mở rộng chức năng mà không ảnh hưởng đến toàn bộ hệ thống, rất phù hợp trong các dự án đang phát triển liên tục hoặc cần tùy biến nhanh.
- *Tối ưu hiệu suất tải:* Thông qua kỹ thuật code-splitting và lazy loading, hệ thống chỉ tải những phần cần thiết, giúp rút ngắn thời gian tải trang ban đầu.
- *Hệ sinh thái phong phú:* React sở hữu cộng đồng lớn cùng nhiều thư viện hỗ trợ mạnh mẽ như React Router (điều hướng), Formik (quản lý form), hay React Query (quản lý truy vấn dữ liệu), góp phần rút ngắn thời gian phát triển.

2.6 Quy trình đóng gói và triển khai với Docker

Để đảm bảo quá trình phát triển và triển khai diễn ra đồng nhất trên mọi máy và môi trường, nhóm lựa chọn sử dụng **Docker** – một công cụ container hóa cho phép đóng gói toàn bộ hệ thống thành các container có thể chạy ở bất cứ đâu.

Một số lợi ích của Docker trong dự án bao gồm:

- *Cài đặt dễ dàng, đồng nhất:* Thay vì phải cấu hình thủ công nhiều công nghệ, Docker giúp nhóm chỉ cần một câu lệnh để khởi tạo toàn bộ hệ thống với đúng phiên bản thư viện và phần mềm đã định trước.
- *Cách ly môi trường làm việc:* Các container hoạt động độc lập với hệ điều hành máy chủ, tránh xung đột cấu hình và giúp bảo vệ hệ thống khỏi những thay đổi ngoài ý muốn.
- *Tránh lỗi do khác biệt môi trường:* Mọi phụ thuộc, bao gồm cả mô hình học máy, thư viện và tệp tin cấu hình, đều được đóng gói cùng nhau, đảm bảo các thành viên đều làm việc với một môi trường thống nhất.
- *Triển khai nhanh chóng:* Khi đã đóng gói, việc triển khai ứng dụng lên server chỉ cần vài lệnh đơn giản, giảm thiểu rủi ro và rút ngắn thời gian đưa sản phẩm vào hoạt động thực tế.

2.7 Kiến trúc hạ tầng triển khai trên AWS

Hệ thống được triển khai trên nền tảng **Amazon Web Services (AWS)** – một trong những nền tảng điện toán đám mây phổ biến nhất hiện nay. Trong đó, nhóm sử dụng chủ yếu hai dịch vụ: **AWS S3** để lưu trữ dữ liệu và **AWS EC2** để chạy ứng dụng backend và mô hình học sâu.

AWS S3 được dùng làm nơi lưu trữ hình ảnh đầu vào. Dịch vụ này cung cấp:

- *Khả năng mở rộng linh hoạt:* Có thể lưu trữ hàng triệu ảnh mà không cần nâng cấp phần cứng hay lo về dung lượng.
- *Tốc độ truy xuất nhanh:* Hình ảnh được truy cập và xử lý gần như ngay lập tức, rất phù hợp với các ứng dụng cần phản hồi theo thời gian thực.
- *Tính bảo mật cao:* AWS S3 hỗ trợ các cơ chế kiểm soát truy cập chi tiết và mã hóa dữ liệu cả khi truyền tải lẫn lưu trữ.
- *Tương thích cao với hệ sinh thái AWS:* Việc tích hợp với các dịch vụ như Lambda, CloudWatch hay EC2 diễn ra trơn tru, giúp mở rộng hệ thống dễ dàng.

AWS EC2 là nơi nhóm triển khai toàn bộ backend và mô hình captioning:

- *Tùy chọn cấu hình phù hợp:* Có thể chọn loại máy ảo theo nhu cầu, từ cấu hình cơ bản đến các máy hỗ trợ GPU để phục vụ huấn luyện và suy diễn mô hình học sâu.
- *Khả năng mở rộng:* Hệ thống có thể mở rộng quy mô tùy theo lượng người dùng thực tế, đảm bảo luôn đáp ứng kịp thời nhu cầu xử lý.
- *Hỗ trợ container:* EC2 tích hợp tốt với Docker, giúp quá trình triển khai diễn ra nhanh chóng và ổn định.
- *Độ tin cậy cao:* Với các chính sách backup, snapshot và recovery của AWS, hệ thống luôn được bảo vệ khỏi các sự cố không mong muốn.

Hệ thống được thiết kế với mục tiêu mang lại khả năng triển khai nhanh chóng và dễ dàng, đặc biệt là với những thành viên trong nhóm chưa có nhiều kinh nghiệp trong lĩnh vực. Với cấu trúc linh hoạt và dễ bảo trì, nhóm có thể nhanh chóng triển khai và điều chỉnh hệ thống khi cần thiết, đảm bảo sự ổn định và hiệu quả trong suốt quá trình phát triển và vận hành.

2.8 Môi trường huấn luyện mô hình học sâu trên nền tảng Kaggle

Thay vì sử dụng các dịch vụ tính phí hoặc tự thiết lập máy chủ GPU, nhóm đã chọn **Kaggle** làm môi trường huấn luyện mô hình do nền tảng này cung cấp GPU miễn phí, ổn định và được tích hợp sẵn các thư viện học sâu phổ biến như TensorFlow, PyTorch, và Keras.

Những lợi ích nổi bật của Kaggle trong dự án:

- *Tiết kiệm chi phí:* Không cần đầu tư phần cứng đắt đỏ hay trả phí dịch vụ cloud, Kaggle đáp ứng đủ nhu cầu xử lý trong giai đoạn huấn luyện.
- *Dễ sử dụng và chia sẻ:* Các notebook có thể dễ dàng chia sẻ giữa các thành viên nhóm mà không cần thiết lập lại môi trường hay tải lại dữ liệu.
- *Hỗ trợ GPU mạnh mẽ:* Giúp rút ngắn đáng kể thời gian huấn luyện mô hình, nhất là khi làm việc với các kiến trúc phức tạp và tập dữ liệu ảnh lớn.

- *Dễ tích hợp vào hệ thống:* Sau khi huấn luyện, mô hình được lưu lại thành các tệp `.json`, `.h5` hoặc `.pt` để phục vụ suy diễn trong ứng dụng thực tế mà không cần huấn luyện lại.
- *Tách biệt giai đoạn huấn luyện và triển khai:* Giúp việc cập nhật mô hình mới nhanh chóng mà không làm gián đoạn hoạt động của hệ thống chính, đảm bảo trải nghiệm người dùng luôn ổn định.

Chương 3

Triển khai hệ thống

Chương này trình bày toàn bộ quá trình triển khai hệ thống, từ thu thập và xử lý dữ liệu đến huấn luyện các mô hình học sâu và triển khai ứng dụng thực tế. Mục tiêu là xây dựng một hệ thống đồng bộ, hiệu quả và có khả năng mở rộng, nhằm giải quyết ba bài toán chính: phân loại ảnh thực phẩm và phi thực phẩm, sinh mô tả hình ảnh, và tạo công thức nấu ăn dựa trên ảnh món ăn. Toàn bộ pipeline bao gồm: thu thập dữ liệu từ nhiều nguồn khác nhau; tổ chức, chuẩn hóa dữ liệu; tinh chỉnh các mô hình học sâu như ResNet50, BLIP, và Image-to-Recipe; và cuối cùng là triển khai hệ thống trên nền tảng đám mây AWS với sự hỗ trợ của Docker và Amazon S3. Kết quả đạt được là bộ dữ liệu quy mô lớn với hơn 225.000 ảnh phục vụ bài toán phân loại, hơn 54.000 ảnh có mô tả đầy đủ, cùng hơn 110.000 công thức nấu ăn. Các mô hình huấn luyện đạt hiệu năng vượt trội: độ chính xác phân loại lên đến 99%, chỉ số ROUGE cao trong sinh mô tả, và F1-score 0.4534 cho bài toán sinh công thức. Hệ thống đã sẵn sàng để triển khai trong thực tế, đồng thời có khả năng mở rộng và tích hợp cho các mục tiêu trong tương lai.

3.1 Thu thập và xử lý dữ liệu

Trong quá trình phát triển hệ thống, một trong những bước quan trọng và tốn nhiều thời gian nhất là xây dựng bộ dữ liệu đầu vào phục vụ cho các mô hình học sâu. Để đảm bảo chất lượng và độ đa dạng, nhóm đã thu thập dữ liệu từ nhiều nguồn đáng tin cậy, bao gồm các kho dữ liệu công khai, trang web ảnh chuyên nghiệp (Pexels, Mixkit, iStockphoto), và các nền tảng chia sẻ công thức nấu ăn như Recipe1M và Food.com.

Các ảnh được xử lý tự động thông qua mô hình phân loại sơ bộ (ResNet50), sau đó được chuẩn hóa về kích thước và định dạng, đồng thời được lưu trữ có tổ chức theo cấu trúc thư mục thống nhất. Các mô tả ảnh và thông tin món ăn được xử lý ngôn ngữ nhằm đảm bảo tính nhất quán, rõ ràng và phù hợp cho việc huấn luyện các mô hình sinh văn bản.Kết quả cuối cùng bao gồm:

- **Bộ dữ liệu phân loại (Food/Non-Food)**: Hơn **225.000 ảnh huấn luyện**, **22.548 ảnh xác thực** và **55.096 ảnh kiểm tra**, phục vụ bài toán phân loại thực phẩm.

- **Bộ dữ liệu mô tả ảnh**: Gồm **54.454 ảnh** từ ba nguồn lớn, trong đó **43.432 ảnh** thuộc lớp `food`, phục vụ cho bài toán *image captioning*.
- **Bộ dữ liệu công thức món ăn**: Kết hợp từ Recipe1M và mở rộng với dữ liệu thức ăn Việt Nam từ Food.com, cung cấp văn bản giàu nội dung cho bài toán sinh công thức từ ảnh.

3.1.1 Dữ liệu phân loại hai lớp

Bài toán phân loại ảnh được thiết kế để xác định xem một ảnh có chứa món ăn hay không. Để huấn luyện mô hình cho tác vụ này, nhóm đã tổng hợp dữ liệu từ nhiều nguồn gồm: `FoodNonFoodDataset.zip`, `combinedsegmentationmodel`, `Caltech-101`, và `Food-5K`. Dữ liệu được tổng hợp và phân chia thành ba tập:

- **Tập huấn luyện**: 225.430 ảnh – dùng để mô hình học các đặc trưng chính.
- **Tập xác thực**: 22.548 ảnh – dùng để theo dõi hiệu quả trong quá trình huấn luyện và tinh chỉnh siêu tham số.
- **Tập kiểm tra**: 55.096 ảnh – dùng để đánh giá khách quan sau khi mô hình đã được huấn luyện.

Tất cả ảnh được resize về kích thước chuẩn **224×224 pixels**, tương thích với kiến trúc đầu vào của ResNet50. Ngoài ra, các kỹ thuật tăng cường dữ liệu như xoay, lật, zoom, và dịch ảnh được áp dụng để cải thiện khả năng tổng quát hoá của mô hình. Dữ liệu được xử lý bằng hàm `preprocess_input` tương thích với mô hình gốc của ResNet50 trong thư viện Keras.

Bộ dữ liệu đã được nén và chia sẻ công khai trên Kaggle: Link tại đây.

3.1.2 Dữ liệu mô tả hình ảnh

Việc thu thập và xử lý dữ liệu mô tả hình ảnh được tiến hành một cách kỹ lưỡng nhằm tạo nên một tập dữ liệu chất lượng cao, phục vụ cho mục tiêu xây dựng mô hình *Image-Captioning*. Bộ dữ liệu này tập trung vào các hình ảnh liên quan đến chủ đề thực phẩm cũng như các ảnh không thuộc chủ đề đó, được tổng hợp từ ba nguồn trực tuyến có uy tín là iStockphoto (`https://www.istockphoto.com/vi`), Mixkit (`https://mixkit.co/`), và Pexels (`https://www.pexels.com/vi-vn/`). Mục đích chính của việc này là tạo ra một nguồn dữ liệu đầu vào phong phú, được phân loại rõ ràng và chuẩn hóa tốt, từ đó hỗ trợ hiệu quả cho các bài toán học sâu liên quan như mô tả hình ảnh, phân loại món ăn hoặc tìm kiếm dựa trên nội dung ngữ nghĩa trong ảnh.

Cụ thể, tổng cộng có 54,454 bức ảnh món ăn được thu thập từ ba trang trên: trong đó iStockphoto đóng góp 47,465 ảnh, Mixkit có 2,201 ảnh và Pexels là 4,788 ảnh, tất cả đều kèm theo phần mô tả chi tiết. Những nguồn dữ liệu này được lựa chọn không chỉ bởi kho ảnh phong phú mà còn bởi chất lượng hình ảnh cao và phần mô tả đi kèm đầy đủ, giúp đảm

bảo độ tin cậy cho việc huấn luyện mô hình. Toàn bộ quá trình thu thập dữ liệu được tự động hóa bằng các công cụ phổ biến trong Python như `Selenium` để tương tác trang web, `BeautifulSoup` để trích xuất dữ liệu HTML, và `Requests` để tải tài nguyên một cách hiệu quả. Thông tin bao gồm hình ảnh và mô tả (thường nằm trong thuộc tính `alt` hoặc phần `Description`) được thu thập đồng thời.

Nguồn Pexels

Đối với nguồn Pexels, dữ liệu được thu thập bằng cách sử dụng thư viện `Selenium` nhằm tự động hóa thao tác cuộn trang và tải về các ảnh liên quan đến từ khóa tìm kiếm `food`.

- *Cấu hình thu thập:* Từ khóa chính được sử dụng là `food`, đồng thời giới hạn số lần cuộn trang để tải ảnh tối đa là 10 lần, được lưu trong biến `SCROLL_LIMIT`. Hình ảnh được lưu trữ vào thư mục `..\pexels\food_images`, và toàn bộ dữ liệu về URL ảnh cũng như phần mô tả được ghi lại trong file `data.csv` với hai cột lần lượt là `Image URL` và `Description`.
- *Quy trình thu thập dữ liệu:* Đầu tiên, chương trình sẽ truy cập vào trang tìm kiếm `https://www.pexels.com/search/food/`, sau đó tự động cuộn trang hoặc nhấn vào nút "Load more" để tải thêm ảnh. Mỗi thẻ `<img>` trên trang sẽ được quét để lấy URL ảnh (thuộc tính `src`) và phần mô tả ảnh (thuộc tính `alt`). Các URL chỉ được giữ lại nếu thuộc miền `images.pexels.com` để đảm bảo độ chính xác, đồng thời tránh trường hợp ảnh trùng lặp bằng cách lưu các URL đã thu thập vào một tập hợp `seen_urls` để kiểm tra. Sau khi lọc, các ảnh sẽ được tải về thư mục cục bộ, còn thông tin về ảnh cùng mô tả sẽ được lưu vào file CSV đã đề cập.
- *Kết quả thu thập:* Tổng cộng đã có 4,788 bức ảnh được tải về, đồng thời thông tin về ảnh và mô tả tương ứng được ghi chép đầy đủ trong file `data.csv` dưới dạng bảng gồm hai cột `[Image URL, Description]`.

Nguồn Mixkit

Để thu thập dữ liệu hình ảnh từ Mixkit, nhóm đã sử dụng các thư viện Python `requests` kết hợp với `BeautifulSoup` nhằm phân tích cấu trúc HTML và tải về các hình ảnh liên quan đến chủ đề thực phẩm.

Phương thức thu thập được thiết kế với bộ từ khóa tìm kiếm đa dạng, bao gồm các danh mục như `food`, `drink`, `coffee`, `tea`, `fruit`, `corn`, `beer`, `cocktail`, `salad`, `eating`, `vegetable`, `fast-food` và `restaurant`. Mỗi danh mục được phép truy cập tối đa 100 trang dữ liệu để đảm bảo đủ số lượng hình ảnh đa dạng. Các ảnh sau khi tải được lưu trữ cục bộ tại thư mục `..\mixkit\image`, đồng thời thông tin chi tiết về ảnh được ghi lại trong file `data.csv`. File này bao gồm các trường dữ liệu như `id` (mã ảnh), `Thumbnail` (đường dẫn ảnh thu nhỏ), `Title` (tiêu đề), `Description` (mô tả) và `status` (trạng thái tải ảnh).

Quá trình thu thập dữ liệu được vận hành theo từng bước như sau: trước hết, hệ thống gửi yêu cầu HTTP đến từng trang của mỗi danh mục, sử dụng thư viện `requests` với cơ chế

tự động thử lại (retry) tối đa 3 lần nếu xảy ra lỗi mạng hoặc kết nối. Để tránh bị khóa truy cập, mỗi yêu cầu được gửi kèm theo một User-Agent được chọn ngẫu nhiên, mô phỏng hành vi của các trình duyệt khác nhau. Dữ liệu HTML nhận về sẽ được phân tích kỹ lưỡng bằng `BeautifulSoup` để trích xuất các thông tin quan trọng gồm tiêu đề, mô tả và ảnh thu nhỏ.

Để tăng tốc độ tải hình ảnh, nhóm triển khai song song quá trình này bằng cách sử dụng `ThreadPoolExecutor` với số lượng worker tối đa lên đến 16, giúp đồng thời xử lý nhiều yêu cầu tải ảnh cùng lúc. Sau khi ảnh được tải về, thông tin từng ảnh sẽ được lưu đồng bộ vào file CSV, đồng thời lưu lại dạng JSON riêng biệt cho từng danh mục (định dạng file `mixkit_category.json`). Bên cạnh đó, nhằm tránh trùng lặp dữ liệu, một tập hợp các đường dẫn ảnh thu nhỏ đã tải (`seen_thumbnails`) được duy trì và kiểm tra liên tục.

Kết quả cuối cùng của quá trình thu thập này là tổng cộng 2,201 hình ảnh duy nhất được tải về và lưu trữ có hệ thống, với dữ liệu chi tiết được ghi lại trong file `mixkit/data.csv` và các file JSON tương ứng, sẵn sàng cho bước xử lý và phân tích tiếp theo của dự án.

Nguồn iStockphoto

Dữ liệu từ iStockphoto được thu thập và xử lý dựa trên các tệp có sẵn, thay vì trực tiếp lấy từ trang web. Cụ thể, thông tin đầu vào bao gồm một file CSV tại đường dẫn `..\istock\data.csv` cùng với thư mục chứa hình ảnh tương ứng tại `..\istock\images`. Mỗi bản ghi trong tập dữ liệu này đều bao gồm các trường quan trọng như `id` – mã định danh duy nhất cho ảnh, `src` – đường dẫn đến file ảnh, `alt` – phần mô tả văn bản thay thế ảnh và `tag` – danh sách các thẻ liên quan phản ánh nội dung hình ảnh. Tổng cộng, sau bước xử lý, nhóm thu thập được 47,465 bức ảnh phục vụ cho các công đoạn phân tích và xử lý tiếp theo.

Cấu trúc dữ liệu chuẩn hóa từ các nguồn

Để thuận tiện cho việc xử lý cũng như phân tích dữ liệu một cách hiệu quả, toàn bộ thông tin thu thập từ các nguồn khác nhau đều được đồng nhất về một cấu trúc chuẩn chung. Mỗi bản ghi dữ liệu sẽ bao gồm những trường quan trọng sau đây, giúp hệ thống dễ dàng quản lý và vận hành:

- **iStockphoto**: Mỗi mục trong bộ dữ liệu này được gán một `id` duy nhất, đóng vai trò như mã nhận dạng để phân biệt các ảnh. Trường `src` lưu lại đường dẫn chính xác tới file ảnh trong hệ thống lưu trữ. Bên cạnh đó, `alt` chứa phần mô tả văn bản thay thế, hỗ trợ cho việc truy vấn dữ liệu và tối ưu hóa công cụ tìm kiếm (SEO). Cuối cùng, `tag` là tập hợp các từ khóa quan trọng, phản ánh nội dung cũng như chủ đề chính của hình ảnh, giúp cho việc lọc và phân loại dữ liệu trở nên linh hoạt hơn.
- **Mixkit**: Dữ liệu từ nguồn này được cấu trúc gồm nhiều trường cần thiết như `id` để nhận dạng duy nhất từng ảnh hoặc video; `Thumbnail` là đường link tới phiên bản ảnh thu nhỏ, giúp tối ưu tốc độ tải khi duyệt danh sách lớn. `Title` đóng vai trò như tiêu đề ngắn gọn,

mang tính tổng quát cho từng mục. `Description` mô tả chi tiết hơn về nội dung của ảnh hoặc video, cung cấp thông tin bổ sung cho người dùng cũng như hệ thống. Ngoài ra, `status` thể hiện trạng thái hiện tại của tệp tin, ví dụ như đã tải xong hay đang trong quá trình xử lý.

- **Pexels**: Với nguồn này, mỗi bản ghi chủ yếu bao gồm hai trường chính. Trường `Image URL` chứa liên kết trực tiếp tới ảnh, giúp dễ dàng truy cập và tải về. Trường `Description` cung cấp một đoạn mô tả ngắn gọn, giúp việc phân loại và tìm kiếm hình ảnh theo nội dung trở nên hiệu quả hơn.

Tiền xử lý dữ liệu

Đổi tên file và chuẩn hóa dữ liệu ảnh từ Pexels:

Để đảm bảo tên file ảnh là duy nhất và dễ quản lý, nhóm tiến hành đổi tên các file ảnh lấy từ Pexels dựa trên URL gốc của chúng. Việc này giúp tránh trùng lặp và đồng thời chuẩn hóa tên file theo một định dạng thống nhất. Cụ thể, nhóm đọc file `data.csv` để lấy thông tin ánh xạ giữa URL ảnh và tên file hiện tại, ví dụ như `image_{index}.jpg` hoặc `.jpeg`.

Qua hàm `get_filename_from_url`, tên file được trích xuất trực tiếp từ URL bằng cách giữ lại chỉ các ký tự chữ, số, dấu gạch ngang và dấu chấm, đồng thời tự động thêm phần mở rộng `.jpg` nếu cần thiết. Sau khi xác định được tên mới, file ảnh trong thư mục `pexels/images` sẽ được đổi tên tương ứng, đồng thời bảng ánh xạ giữa tên file cũ và tên file mới cũng được lưu lại. Cuối cùng, một file mới có tên `processed.csv` được tạo ra, bao gồm các trường dữ liệu như `id`, `src` (đường dẫn ảnh), và `alt` (thẻ mô tả ảnh).

Phân loại ảnh theo nguồn dữ liệu: Pexels, Mixkit và iStock

Để phân nhóm các ảnh thành hai loại chính là `food` (mã 0) và `nonfood` (mã 1), nhóm sử dụng mô hình deep learning `ResNet50` đã được tinh chỉnh và huấn luyện trước đó. Mô hình này nhận ảnh đầu vào, thực hiện dự đoán và phân loại ảnh thành từng nhóm phù hợp. Sau khi dự đoán, ảnh được di chuyển vào các thư mục tương ứng theo kết quả phân loại, giúp dễ dàng quản lý và truy cập.

Cấu hình chi tiết của hệ thống xử lý:

- **Mô hình được sử dụng:** `best_resnet_model.h5`
- **Cấu trúc thư mục đầu vào - đầu ra theo từng nguồn:**
  - **Pexels:** ảnh gốc trong `pexels/images`, kết quả phân loại lưu trong `pexels/sorted/food` và `pexels/sorted/nonfood`.
  - **Mixkit:** ảnh trong `mixkit/images` được sắp xếp vào `mixkit/sorted/food` và `mixkit/sorted/nonfood`.
  - **iStock:** ảnh gốc ở `istock/images`, phân loại vào `istock/sorted/food` và `istock/sorted/nonfood`.

- **File kết quả tổng hợp:** `data_with_class.csv`, bao gồm:
  - Cột `class`: nhãn phân loại (0 hoặc 1).
  - Cột `confidence`: độ tin cậy của dự đoán.
- **Kích thước batch xử lý:** 64 ảnh (giúp tối ưu tốc độ và bộ nhớ).

Quy trình xử lý ảnh

Ảnh đầu vào được đọc và chuyển đổi về kích thước tiêu chuẩn 224x224 pixel, chuyển sang định dạng màu RGB để phù hợp với yêu cầu của mô hình ResNet50. Tiếp theo, ảnh được chuẩn hóa bằng hàm `preprocess_input` đặc thù, giúp dữ liệu phù hợp với kiến trúc mạng và nâng cao hiệu quả dự đoán.

Việc dự đoán được thực hiện theo từng lô ảnh (batch) để tối ưu tài nguyên. Đồng thời, nhóm áp dụng hai kỹ thuật nâng cao nhằm tăng tốc và đảm bảo hiệu suất xử lý:

- Sử dụng `ThreadPoolExecutor` để song song hóa quá trình tải và sao chép ảnh, từ đó rút ngắn thời gian chờ đợi.
- Áp dụng kỹ thuật `mixed_float16` nhằm giảm thiểu mức tiêu thụ bộ nhớ và đẩy nhanh quá trình tính toán mà không làm giảm độ chính xác.

Các file ảnh không tìm thấy hoặc có lỗi trong quá trình đọc được ghi lại vào file `missing_files.csv`, giúp thuận tiện cho việc kiểm tra và xử lý lại sau này.

Kết quả cuối cùng là các ảnh được phân loại chính xác, lưu vào thư mục `sorted/food` hoặc `sorted/nonfood`. Đồng thời, thông tin phân loại chi tiết được tổng hợp trong `data_with_class.csv` để phục vụ cho bước phân tích và đánh giá chất lượng dữ liệu. Trong toàn bộ quá trình, có tổng cộng 43,432 ảnh được xác định là thuộc nhóm `food`.

Chuẩn hóa và hợp nhất dữ liệu

Ba nguồn dữ liệu với cấu trúc khác nhau được xử lý và chuẩn hóa nhằm tạo ra một tập dữ liệu đồng nhất, thuận tiện cho việc huấn luyện mô hình. Các bước cụ thể trong quá trình này được thực hiện như sau:

- **iStockphoto:** Trước tiên, cột `tag` không phục vụ mục đích phân loại nên được loại bỏ để giảm bớt dữ liệu không cần thiết. Đồng thời, kiểm tra lại cột `id` để bổ sung phần đuôi `.jpg` nếu chưa có, nhằm đảm bảo các tên tệp ảnh có định dạng chuẩn.
- **Mixkit:** Với dữ liệu từ Mixkit, trường `src` được tạo mới dựa trên giá trị trong cột `Thumbnail` để dễ dàng truy cập hình ảnh. Phần mô tả thay thế `alt` được sinh ra ưu tiên từ cột `Description`, nếu trường này trống thì thay thế bằng `Title`. Ngoài ra, những cột không cần thiết như `status`, `Title`, `Description` cũng được loại bỏ nhằm giữ lại các thông tin quan trọng nhất.

- **Pexels:** Dữ liệu đã được xử lý trước đó và lưu trong tệp `processed.csv` được dùng trực tiếp, đảm bảo tính nhất quán với quy trình chuẩn hóa chung.

Sau khi thực hiện chuẩn hóa riêng biệt, ba bộ dữ liệu được gom lại thành một bảng dữ liệu duy nhất, lưu trữ trong file `data_processed.csv`. Bộ dữ liệu này gồm các cột quan trọng như `id` (tên ảnh), `src` (đường dẫn ảnh), `alt` (mô tả thay thế), `class` (nhóm ảnh food hoặc nonfood) và `confidence` (độ tin cậy nhãn).

Tiếp đó, toàn bộ ảnh thuộc thư mục `sorted/food` và `sorted/nonfood` được sao chép lại và tổ chức lại trong hai thư mục thống nhất là `all_images/food` và `all_images/nonfood`. Để tránh trùng tên khi gộp ảnh từ nhiều nguồn khác nhau, mỗi tệp ảnh được đặt tên mới với tiền tố tương ứng nguồn dữ liệu: `istock_`, `pexels_`, hoặc `mixkit_`, kèm theo một số thứ tự để đảm bảo tính duy nhất. Tên ảnh mới này được ghi lại trong cột `new_filename` của DataFrame.

Kết quả cuối cùng được lưu thành file `data_processed_updated.csv`, chứa đầy đủ các thông tin bao gồm: `id`, `src`, `alt`, `class`, `confidence` và `new_filename`. Bộ dữ liệu này tổng cộng có **54,454** ảnh, trong đó riêng nhóm ảnh `food` chiếm số lượng lớn với **43,432** ảnh.

Bộ dữ liệu hoàn chỉnh đã được đóng gói và đăng tải trên Kaggle có thể truy cập tại đây. [1].

Như vậy, tập dữ liệu cuối cùng bao gồm 54,454 ảnh được sắp xếp trong hai thư mục riêng biệt cho ảnh `food` và `nonfood`. File `data_processed_updated.csv` đóng vai trò là bảng metadata tổng hợp, cung cấp thông tin về tên ảnh gốc, đường dẫn, mô tả thay thế, nhãn lớp, mức độ tin cậy và tên tệp ảnh sau khi đổi tên (xem minh họa tại Bảng 3.1).

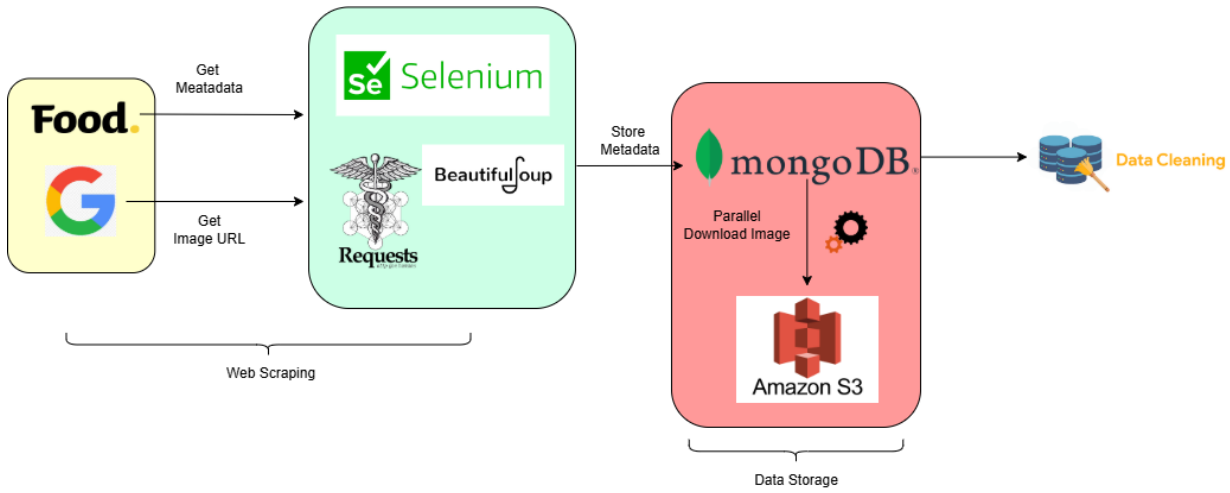
ID	URL (src)	Alt Text	Class	Confidence	Tên Mới
b760a522-...0ea1d3a.jpg	`https://media.istockphoto.com/id/450752471/...`	homemade organic dessert ready to eat	0	0.99999976	istock_b760a522-82de-4fea-b43c-9985f0ea1d3a.jpg
84fc5fcf-...a45a636.jpg	`https://media.istockphoto.com/id/184350005/...`	isolated white background a slice of on a plate clipping path included	0	1.0	istock_84fc5fcf-1d02-41f4-812d-acea7a45a636.jpg

Bảng 3.1: Mẫu dữ liệu từ `data_processed_updated.csv`

Tập dữ liệu này hiện đã được chuẩn bị đầy đủ, sẵn sàng cho các bước tiếp theo như huấn luyện mô hình tạo caption cho ảnh.

[1] `https://www.kaggle.com/datasets/nguyenngoc7777/food-caption-dataset/`

3.1.3 Dữ liệu thông tin món ăn

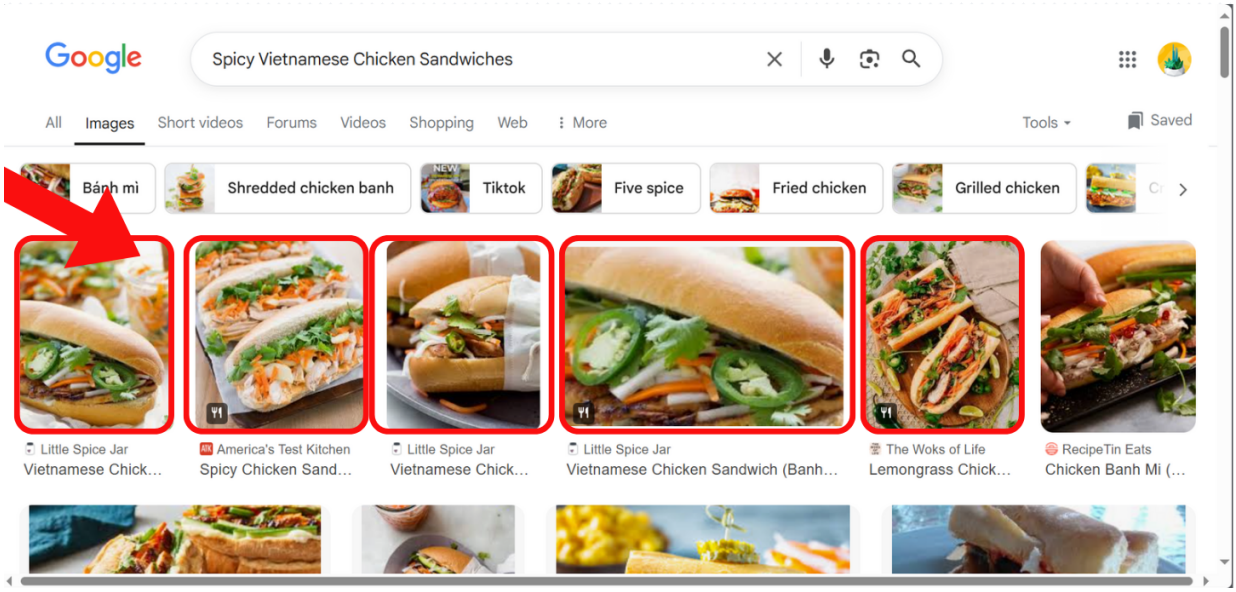


Hình 3.1: Minh họa quy trình thu thập dữ liệu

Dự án sử dụng một phần của bộ dữ liệu **Recipe1M** – một nguồn dữ liệu phong phú chứa hơn 110.000 công thức nấu ăn thu thập từ nhiều trang web ẩm thực uy tín. Mỗi bản ghi trong tập dữ liệu bao gồm các thành phần chính: tiêu đề món ăn, danh sách nguyên liệu, hướng dẫn chế biến chi tiết, cùng với liên kết dẫn về trang web gốc. Qua bước lọc sơ bộ nhằm loại bỏ các công thức thiếu thông tin hoặc không đi kèm hình ảnh minh họa, bộ dữ liệu cuối cùng được chia thành ba phần riêng biệt gồm: tập huấn luyện khoảng 78.000 món, tập xác thực gần 17.000 món, và tập kiểm tra cũng với khoảng 17.000 món ăn.

Tuy nhiên, trong quá trình rà soát nhanh dữ liệu, nhóm phát hiện số lượng món ăn Việt Nam trong tập này khá hạn chế, chưa đến 100 bản ghi có liên quan đến món Việt. Để khắc phục điểm này, nhóm đã tiến hành bổ sung thêm dữ liệu từ trang Food.com, một nguồn cung cấp công thức nấu ăn khá đa dạng và tương thích về mặt cấu trúc dữ liệu, thu được khoảng 800 món ăn Việt cùng với tên gọi và liên kết đến trang gốc.

Với số lượng món Việt còn khá khiêm tốn, nhóm tiếp tục thu thập thêm hình ảnh minh họa liên quan bằng cách sử dụng các từ khóa tìm kiếm tương ứng trên Google Images. Mỗi từ khóa sẽ được lấy 5 hình ảnh đầu tiên nhằm đảm bảo tính liên quan và độ chính xác cao. Do ảnh hiển thị trên Google thường dưới dạng thumbnail với kích thước nhỏ, không đáp ứng được yêu cầu về chất lượng cho bài toán, nhóm đã tiến hành thu thập các đường dẫn mở rộng của 5 vị trí đầu tiên này rồi mới tải về ảnh với kích thước gốc, đảm bảo dữ liệu đầu vào có độ phân giải phù hợp.



Hình 3.2: Ảnh minh họa tìm kiếm bằng từ khóa trên Google Images

Vì tổng số ảnh cần xử lý tương đối lớn, nhóm không thực hiện tải về tất cả ảnh một cách tuần tự mà chỉ thu thập liên kết ảnh trước. Tiếp đó, quá trình tải ảnh được thực hiện song song sử dụng thư viện **joblib**, vừa giúp tăng tốc độ vừa tạo ID duy nhất đồng bộ trong lúc tải. Ảnh tải về sau đó được lưu trữ trên dịch vụ AWS S3 nhằm đảm bảo tính bền vững dữ liệu, phòng trường hợp đường dẫn gốc có thể thay đổi hoặc bị xóa trong tương lai.

Dữ liệu thu thập được tổ chức lại dưới dạng các trường thông tin chính, bao gồm:

- **title:** Tên món ăn.
- **directions:** Hướng dẫn chi tiết cách chế biến món ăn.
- **NER:** Danh sách nguyên liệu chính sử dụng trong công thức.
- **link:** Đường dẫn tới trang web gốc của công thức.
- **image_url:** Đường dẫn tới ảnh minh họa món ăn.
- **id:** Mã định danh được bổ sung tương ứng khi ảnh được tải về và lưu trữ.

Trước khi chuyển sang các bước xử lý và mô hình hóa tiếp theo, dữ liệu được tiền xử lý cẩn thận để đảm bảo chất lượng và phù hợp với yêu cầu. Trong đó, hệ thống xử lý song song hai loại dữ liệu chính:

- **Hình ảnh:** Bao gồm ảnh chụp món ăn với đa dạng độ phân giải và định dạng khác nhau, đòi hỏi các bước chuẩn hóa phù hợp để sử dụng trong bài toán học sâu.
- **Văn bản mô tả:** Gồm tiêu đề món ăn, danh sách nguyên liệu và hướng dẫn chế biến – những thông tin này cần được xử lý ngôn ngữ tự nhiên để trích xuất tri thức hữu ích phục vụ cho việc chú thích ảnh.

Tiền xử lý phần mô tả văn bản

Để đảm bảo dữ liệu văn bản đầu vào có định dạng đồng nhất, dễ hiểu và phù hợp với mô hình học máy, quá trình tiền xử lý mô tả được thực hiện theo nhiều bước cụ thể như sau:

- **Làm sạch tên món ăn:**

  Với những tiêu đề món ăn chứa hai phần song song trong dấu ngoặc (ví dụ: *Vietnamese Caramelized Salty Pork (Thịt Kho)*), ta sẽ áp dụng quy tắc chọn lọc dựa trên ngôn ngữ và ngữ nghĩa. Nếu tiêu đề thuộc về món ăn Việt Nam, phần tên tiếng Việt sẽ được ưu tiên giữ lại. Ngược lại, nếu món ăn mang tính quốc tế, sẽ chọn phần nào dài hơn để giữ thông tin đầy đủ. Thư viện `langdetect` được sử dụng để tự động nhận diện ngôn ngữ nhằm hỗ trợ cho bước lựa chọn này. Ngoài ra, mọi tiêu đề có chứa cụm từ "Vietnamese" đều được ưu tiên giữ nguyên để làm rõ nguồn gốc món ăn. Việc này giúp hệ thống nhận diện và phân loại dữ liệu hiệu quả hơn về mặt ngữ cảnh văn hóa và ẩm thực.

- **Xử lý văn bản hướng dẫn nấu ăn:**

  Các công thức thường được trình bày dưới dạng liệt kê theo từng bước (ví dụ: "1. Mix the sauce...", hoặc "STEP 2: Heat the oil..."). Những định dạng này có thể gây nhiễu cho mô hình ngôn ngữ, vì vậy nhóm đã sử dụng biểu thức chính quy để loại bỏ số thứ tự và các cụm từ mở đầu không cần thiết, đồng thời nối toàn bộ phần hướng dẫn thành một đoạn văn thống nhất, liền mạch và dễ học đối với mô hình.

- **Giải mã các ký tự đặc biệt:**

  Trong quá trình thu thập dữ liệu từ nhiều nguồn trên web, một số ký tự đặc biệt bị mã hóa theo chuẩn HTML (ví dụ: `&`, `<`, `>`, `&quot;`). Những ký tự này được giải mã trở lại đúng biểu diễn gốc để đảm bảo văn bản không bị sai lệch hoặc khó hiểu.

- **Chuẩn hóa các ký hiệu viết tắt:**

  Để giảm sự đa dạng không cần thiết trong cách biểu diễn, các ký hiệu phổ biến như `'N`, `N'`, `&`... đều được chuyển đổi thành dạng chuẩn là "AND". Điều này giúp làm giảm số lượng từ vựng khác biệt không cần thiết và tăng độ đồng nhất dữ liệu.

- **Đưa toàn bộ văn bản về dạng chữ thường:**

  Việc đồng nhất chữ hoa và chữ thường là một bước quan trọng trong tiền xử lý dữ liệu ngôn ngữ. Toàn bộ phần mô tả được chuyển về chữ thường để tránh trùng lặp không cần thiết (ví dụ: "Sugar" và "sugar" được xem là cùng một từ).

Nhờ vào những bước xử lý này, dữ liệu đầu vào được chuẩn hóa và đơn giản hóa, giúp mô hình học hiệu quả hơn, tránh bị ảnh hưởng bởi sự nhiễu loạn trong cách biểu diễn ngôn ngữ ban đầu.

Tiền xử lý dữ liệu hình ảnh

Song song với việc xử lý văn bản, dữ liệu ảnh cũng được xử lý kỹ càng để đảm bảo đầu vào cho mô hình thị giác máy tính đạt chất lượng cao và nhất quán:

- **Loại bỏ ảnh chất lượng thấp:**

  Những hình ảnh có kích thước quá nhỏ (chiều rộng hoặc chiều cao đều dưới 256 pixel) sẽ không được sử dụng. Việc phóng lớn những ảnh này để đạt kích thước yêu cầu không những không cải thiện mà còn làm giảm chất lượng ảnh (bị mờ, vỡ), từ đó ảnh hưởng tiêu cực đến khả năng học đặc trưng thị giác của mô hình.

- **Thay đổi kích thước ảnh lớn:**

  Với những hình ảnh có kích thước lớn hơn, ảnh sẽ được resize sao cho chiều ngắn nhất đạt đúng 256 pixel. Tỷ lệ gốc của ảnh sẽ được giữ nguyên nhằm tránh méo hình, đảm bảo cấu trúc hình ảnh vẫn tự nhiên.

- **Chuẩn hóa giá trị pixel:**

  Cuối cùng, toàn bộ ảnh được chuẩn hóa pixel theo giá trị trung bình và độ lệch chuẩn của bộ dữ liệu ImageNet. Điều này giúp mô hình tiền huấn luyện (pretrained model) như ResNet, EfficientNet hoặc Inception hoạt động hiệu quả hơn khi fine-tune trên bộ dữ liệu của hệ thống, do đầu vào đã phù hợp với kỳ vọng ban đầu của mô hình.

Với quy trình tiền xử lý như trên, dữ liệu hình ảnh đầu vào được đảm bảo chất lượng, nhất quán về kích thước và phân bố pixel, từ đó hỗ trợ tốt hơn cho quá trình huấn luyện và suy luận của mô hình nhận dạng và sinh mô tả ảnh.

3.2 Triển khai các mô hình

Nhóm dự án triển khai ba mô hình học sâu nhằm giải quyết các bước chính của hệ thống: ResNet50 để phân loại ảnh Food/Non-food, BLIP để sinh mô tả ảnh món ăn, và Image-to-Recipe để tạo công thức nấu ăn từ ảnh. Với ResNet50, nhóm giữ nguyên các tầng đầu nhằm khai thác đặc trưng cơ bản và chỉ tinh chỉnh 30 tầng cuối, đạt độ chính xác 98.17% trên tập xác thực và 99% trên tập kiểm tra. BLIP – dựa trên kiến trúc Transformer – được tinh chỉnh trên 10.000 ảnh từ Food Caption Dataset, giảm mất mát từ 5.4667 xuống 1.8456 sau 10 epoch, sinh caption tự nhiên, đánh giá bằng BLEU, METEOR và ROUGE. Image-to-Recipe kết hợp ResNet50 và Transformer Decoder, huấn luyện trên Recipe1M và dữ liệu món Việt, đạt 98.74% độ chính xác nhưng F1 Ingredients còn thấp (0.0652), do món Việt có nhiều thành phần khó nhận diện. Quá trình huấn luyện sử dụng tăng cường dữ liệu, EarlyStopping và ReduceLROnPlateau để tối ưu. Kết quả cho thấy hệ thống hoạt động hiệu quả, nhưng cần tiếp tục cải thiện khả năng nhận diện nguyên liệu, đặc biệt với món ăn Việt Nam, bằng cách mở rộng dữ liệu và tối ưu mô hình.

3.2.1 Khái niệm về tinh chỉnh mô hình (Finetuning)

Tinh chỉnh (finetuning) là một kỹ thuật phổ biến trong học sâu, đặc biệt hữu ích khi ta muốn tận dụng sức mạnh của các mô hình đã được huấn luyện trước trên những bộ dữ liệu lớn và đa dạng, thay vì bắt đầu huấn luyện từ con số không. Ý tưởng cốt lõi là kế thừa những gì mô hình đã "học" được — ví dụ như cách nhận diện đường nét, hình khối, màu sắc hay các mẫu đặc trưng trong hình ảnh — và chỉ điều chỉnh lại một phần của mạng để thích ứng với nhiệm vụ cụ thể của chúng ta.

Trong trường hợp này, nhóm đã sử dụng các mô hình học sâu nổi tiếng như ResNet50 cho bài toán phân loại ảnh (chẳng hạn xác định ảnh đó có phải là món ăn hay không), BLIP để sinh mô tả (caption) cho ảnh món ăn, và một mô hình Image-to-Recipe để tự động tạo ra công thức nấu ăn tương ứng từ hình ảnh.

Thay vì huấn luyện toàn bộ mạng nơ-ron từ đầu — điều này thường đòi hỏi nhiều tài nguyên tính toán và dữ liệu lớn — chúng tôi lựa chọn chiến lược sau:

- **Giữ lại các tầng đầu của mô hình gốc**: Những tầng này thường học được các đặc trưng cơ bản như mép ảnh, hình dạng tổng quát, hoặc các đặc điểm màu sắc. Vì các đặc trưng này tương đối phổ quát, chúng có thể tái sử dụng cho nhiều bài toán khác nhau mà không cần điều chỉnh nhiều.
- **Điều chỉnh lại các tầng cuối**: Đây là phần mà mô hình học cách "hiểu" bài toán cụ thể. Chúng tôi thay thế các tầng phân loại gốc bằng các lớp phù hợp hơn với nhiệm vụ cần giải — ví dụ như phân loại ảnh thành hai nhóm "Food"hoặc "Non-food", hoặc sinh mô tả bằng văn bản.

Chiến lược này giúp rút ngắn thời gian huấn luyện, tiết kiệm tài nguyên phần cứng, đồng thời vẫn đạt được hiệu quả tốt trên tập dữ liệu mục tiêu.

Các mô hình đã được tinh chỉnh trong hệ thống gồm có:

- **ResNet50** cho nhiệm vụ nhận diện và phân loại ảnh món ăn.
- **BLIP (Bootstrapped Language-Image Pretraining)** để tự động sinh chú thích cho ảnh món ăn.
- **Image-to-Recipe Model** dùng để sinh công thức nấu ăn dựa trên nội dung hình ảnh.

Việc tích hợp và tinh chỉnh các mô hình này là bước then chốt, giúp hệ thống có khả năng xử lý và phản hồi hiệu quả trước các tác vụ đa dạng liên quan đến dữ liệu hình ảnh thực tế mà người dùng cung cấp.

3.2.2 Mô hình phân loại hình ảnh - ResNet50

Để thực hiện nhiệm vụ phân loại hình ảnh trong hệ thống, nhóm đã lựa chọn sử dụng kiến trúc **ResNet50**, một biến thể nổi bật trong dòng mô hình mạng nơ-ron tích chập sâu (CNN

– Convolutional Neural Network). ResNet (viết tắt của Residual Network) lần đầu được giới thiệu vào năm 2015 bởi nhóm nghiên cứu tại Microsoft Research và nhanh chóng tạo dấu ấn khi đạt kết quả xuất sắc tại các cuộc thi lớn như ILSVRC và COCO trong cùng năm. Ưu điểm lớn nhất của kiến trúc này là khả năng xử lý hiệu quả độ sâu lớn mà không gặp phải hiện tượng mất mát gradient – một vấn đề phổ biến khi huấn luyện các mạng nhiều tầng.

Mô hình ResNet50 được đặt tên dựa trên số lượng lớp học được trong mạng – tổng cộng có 50 lớp bao gồm nhiều khối residual (bổ trợ) cho phép luồng gradient được truyền ngược dễ dàng hơn trong quá trình huấn luyện. Mỗi khối residual hoạt động như một "cầu nối" giữa đầu vào và đầu ra của một nhóm tầng, từ đó giúp mô hình học các đặc trưng phức tạp mà không làm suy giảm chất lượng học theo chiều sâu.

Việc lựa chọn ResNet50 là kết quả của sự cân nhắc kỹ lưỡng giữa hiệu quả và chi phí tính toán. Mô hình này đáp ứng tốt các yêu cầu của bài toán trong bối cảnh bộ dữ liệu không quá lớn — chẳng hạn như tập dữ liệu phân loại ảnh Food vs. Non-Food chỉ bao gồm vài chục nghìn ảnh. Trong khi đó, các kiến trúc hiện đại hơn như Vision Transformer (ViT) tuy có nhiều tiềm năng nhưng lại đòi hỏi lượng dữ liệu và tài nguyên huấn luyện lớn hơn nhiều để hoạt động ổn định.

Các lý do chính khiến nhóm lựa chọn ResNet50 bao gồm:

- **Hiệu quả trong trích xuất đặc trưng:** Khả năng học được các đặc trưng hình ảnh từ mức độ cơ bản (low-level) như cạnh và họa tiết đến mức độ trừu tượng cao hơn (high-level) như hình dạng và đối tượng.
- **Tương thích với quy mô dữ liệu trung bình:** Với kích thước tập dữ liệu không quá lớn, ResNet50 vẫn cho kết quả tốt mà không yêu cầu huấn luyện từ đầu hoặc sử dụng tài nguyên vượt quá mức cho phép.
- **Tính ổn định và khả năng mở rộng:** Cấu trúc residual giúp khắc phục tình trạng gradient vanishing, tạo nền tảng vững chắc cho huấn luyện mô hình sâu, đồng thời vẫn có thể mở rộng hoặc fine-tune linh hoạt khi cần.

Trong thiết kế mô hình, tầng đầu ra được xây dựng với một nút duy nhất sử dụng hàm kích hoạt **sigmoid**, nhằm đưa ra xác suất dự đoán hình ảnh thuộc lớp **Food**. Cụ thể:

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

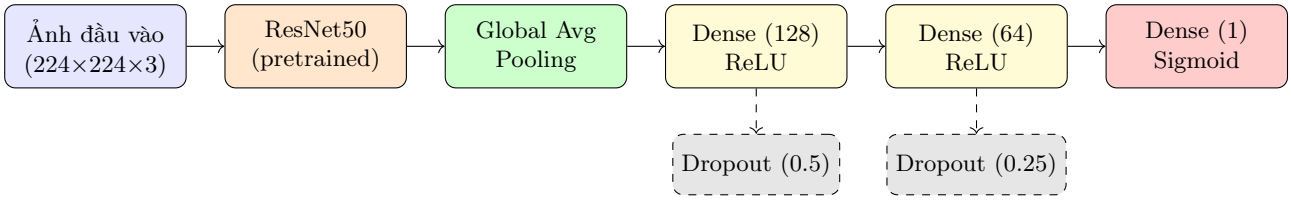
Trong đó, z là giá trị đầu ra từ lớp Dense cuối cùng của mô hình. Giá trị $\hat{y}$ càng gần 1 thì khả năng ảnh thuộc lớp "Food"càng cao, ngược lại càng gần 0 thì có xu hướng là "Non-Food".

Quá trình huấn luyện được tối ưu dựa trên hàm mất mát binary cross-entropy, vốn là tiêu chuẩn cho các bài toán phân loại nhị phân:

$$\mathcal{L}(y, \hat{y}) = -\left[y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y})\right]$$

Trong đó, y là nhãn thực tế (0 hoặc 1) và $\hat{y}$ là xác suất dự đoán từ mô hình. Hàm mất mát này đo lường mức độ sai lệch giữa dự đoán và thực tế, từ đó hướng dẫn mô hình điều chỉnh trọng số trong quá trình huấn luyện.

Tóm lại, ResNet50 không chỉ phù hợp về mặt kỹ thuật mà còn đảm bảo cân bằng giữa độ chính xác, hiệu năng và yêu cầu tài nguyên, trở thành lựa chọn hợp lý cho bài toán phân loại ảnh trong dự án này.



Hình 3.3: Kiến trúc mô hình phân loại ResNet50

Quy trình xử lý ảnh trong mô hình phân loại

Toàn bộ quy trình xử lý ảnh trong mô hình được thiết kế theo hướng tối ưu hiệu quả trích xuất đặc trưng và tăng khả năng tổng quát hóa trong quá trình huấn luyện. Dưới đây là các bước chính trong pipeline xử lý ảnh:

1. **Tiền xử lý đầu vào**: Hình ảnh đầu vào được chuẩn hóa và điều chỉnh về kích thước cố định 224×224 pixel, với 3 kênh màu RGB. Việc chuẩn hóa kích thước không chỉ giúp đảm bảo tính đồng nhất cho đầu vào mà còn phù hợp với yêu cầu của kiến trúc ResNet50.

2. **Trích xuất đặc trưng với ResNet50**: Hình ảnh sau khi được chuẩn hóa sẽ được đưa vào mô hình ResNet50 – một mạng nơ-ron tích chập sâu được huấn luyện sẵn trên tập dữ liệu ImageNet. Trong quá trình này, phần lớn các lớp ban đầu của mô hình được giữ nguyên để tận dụng các đặc trưng tổng quát đã học từ ImageNet, trong khi một số lớp cuối được tinh chỉnh (fine-tuned) nhằm thích nghi với bài toán cụ thể hiện tại.

3. **Lớp Pooling toàn cục (Global Average Pooling)**: Sau bước trích xuất đặc trưng, mô hình sử dụng kỹ thuật pooling toàn cục để tóm tắt toàn bộ bản đồ đặc trưng thành một vector duy nhất. Việc này giúp giảm mạnh số lượng tham số, đồng thời vẫn giữ lại thông tin quan trọng của ảnh.

4. **Lớp ẩn Dense 128 đơn vị (ReLU)**: Đầu ra từ bước pooling được đưa vào một lớp fully connected gồm 128 nút kích hoạt bằng hàm ReLU. Mục tiêu của lớp này là học các đặc trưng phân biệt có ý nghĩa hơn cho bài toán phân loại.

5. **Dropout (tỉ lệ 0.5)**: Để hạn chế hiện tượng học quá khớp (overfitting), một lớp dropout với tỉ lệ 50% được chèn vào. Trong mỗi lần huấn luyện, lớp này ngẫu nhiên "vô hiệu hóa" một nửa số nút thần kinh, giúp mô hình học được các đặc trưng ổn định hơn.

6. **Lớp ẩn Dense 64 đơn vị (ReLU)**: Một lớp fully connected tiếp theo với 64 nút tiếp tục xử lý và khai thác chiều sâu của biểu diễn đặc trưng, cho phép mô hình học thêm các mối quan hệ phức tạp hơn giữa các thông tin đã trích xuất.

7. **Dropout (tỉ lệ 0.25)**: Tiếp tục chèn một lớp dropout với tỉ lệ thấp hơn (25%) để duy trì khả năng khái quát, đồng thời tránh việc mô hình trở nên quá phụ thuộc vào các nút cụ thể.

8. **Lớp đầu ra Dense 1 đơn vị (Sigmoid)**: Cuối cùng, mô hình kết thúc bằng một lớp đầu ra với một nút duy nhất sử dụng hàm kích hoạt sigmoid. Kết quả trả về là một giá trị xác suất nằm trong khoảng [0, 1], đại diện cho mức độ tự tin của mô hình khi phân loại ảnh đầu vào thuộc về lớp "Food" hoặc "Non-Food".

Chiến lược tinh chỉnh và huấn luyện

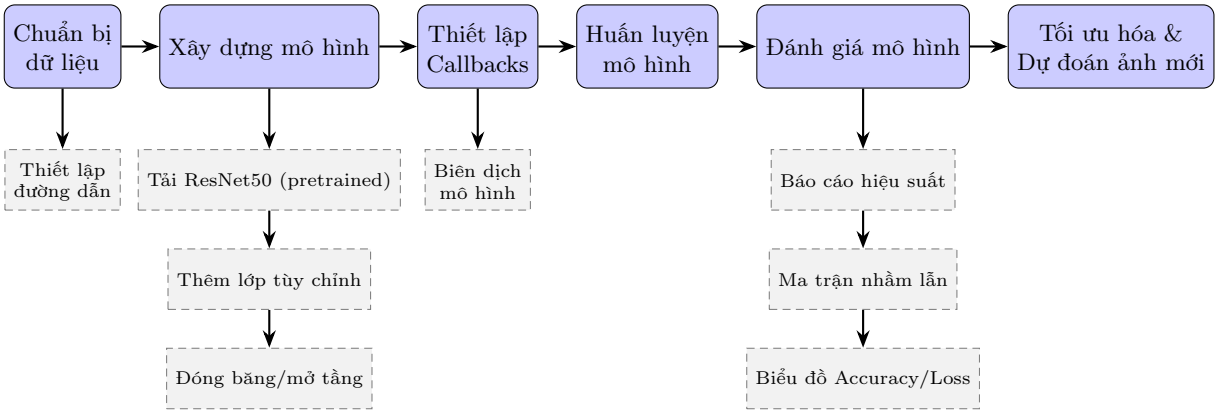
Để tăng khả năng thích nghi của mô hình với bài toán cụ thể trong dự án, nhóm thực hiện tinh chỉnh mô hình ResNet50 bằng cách khóa (đóng băng) toàn bộ các lớp của mạng nền, ngoại trừ 30 lớp cuối cùng. Cách tiếp cận này cho phép bảo toàn các đặc trưng tổng quát đã học từ tập dữ liệu ImageNet, đồng thời cho phép các tầng sâu hơn được điều chỉnh để phù hợp hơn với dữ liệu hình ảnh thực tế trong bài toán phân loại hiện tại. Việc tối ưu hóa mô hình được thực hiện bằng thuật toán Adam, với tốc độ học khởi điểm được đặt ở mức tương đối nhỏ là 1×10^{-4} nhằm đảm bảo quá trình huấn luyện diễn ra ổn định.

Trong suốt quá trình huấn luyện, mô hình sử dụng hàm mất mát *binary cross-entropy*, vốn là lựa chọn phù hợp cho các bài toán phân loại nhị phân như phân biệt giữa ảnh đồ ăn và không phải đồ ăn. Độ chính xác (accuracy) được sử dụng như chỉ số chính để theo dõi và đánh giá hiệu suất mô hình trên cả tập huấn luyện lẫn tập xác thực.

Để giúp mô hình thích nghi tốt hơn trong những giai đoạn sau của quá trình huấn luyện, nhóm áp dụng hàm gọi lại `ReduceLROnPlateau`. Cơ chế này giúp tự động giảm tốc độ học xuống còn một nửa nếu hàm mất mát trên tập xác thực không được cải thiện sau hai epoch liên tiếp. Đồng thời, nhóm cũng tích hợp `EarlyStopping` — một cơ chế dừng sớm khi nhận thấy không có cải thiện rõ rệt về hiệu suất trong ba epoch liên tục, nhằm tránh tình trạng huấn luyện dư thừa hoặc dẫn đến quá khớp (overfitting).

Mô hình được huấn luyện trong môi trường sử dụng GPU với kích thước lô (batch size) là 32. Trong thực tế, quá trình huấn luyện diễn ra khá nhanh và ổn định, tận dụng tốt sức mạnh tính toán của phần cứng và khả năng học chuyển tiếp (transfer learning) từ mô hình tiền huấn luyện.

Dưới đây là sơ đồ mô tả toàn bộ quy trình tinh chỉnh ResNet50 mà nhóm đã triển khai:



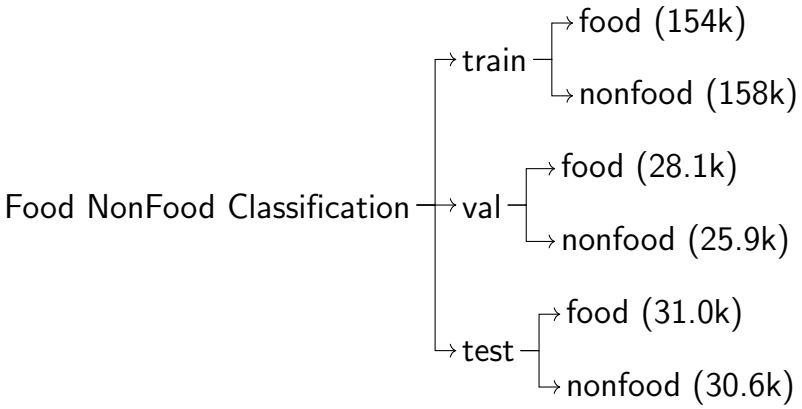
Hình 3.4: Sơ đồ quy trình tinh chỉnh ResNet50

Sơ đồ trên phản ánh một luồng quy trình tuyến tính, bắt đầu từ giai đoạn chuẩn bị dữ liệu đầu vào, tiếp nối bởi việc xây dựng và tinh chỉnh kiến trúc mô hình, cho đến huấn luyện, đánh giá và tối ưu hóa để phục vụ cho dự đoán trên các ảnh chưa từng thấy trước đó. Việc tổ chức quy trình này giúp đảm bảo tính logic, kiểm soát hiệu quả hiệu suất mô hình và dễ dàng tái sử dụng trong các dự án tương tự về sau.

Chi tiết quy trình tinh chỉnh mô hình Resnet50

Bước 1: Chuẩn bị dữ liệu đầu vào

- *Thiết lập đường dẫn dữ liệu:* Xác định các thư mục chứa dữ liệu từ bộ dữ liệu *Food NonFood Classification Dataset* trên Kaggle. Bộ dữ liệu được tổ chức thành ba thư mục. Mỗi thư mục được chia thành hai thư mục con: `Food` (ảnh thực phẩm, như pizza, hamburger) và `Non-food` (ảnh không phải thực phẩm, như xe hơi, bàn ghế).



Hình 3.5: Cấu trúc thư mục chứa dữ liệu huấn luyện, kiểm thử và xác thực.

- *Tăng cường dữ liệu (Data Augmentation):* Để cải thiện khả năng tổng quát hóa và tránh hiện tượng học thuộc (overfitting), áp dụng các biến đổi ngẫu nhiên cho ảnh trong tập huấn luyện, bao gồm:
  - *Xoay ảnh:* Xoay tối đa 15 độ để mô phỏng các góc nhìn khác nhau.

- *Dịch chuyển:* Di chuyển ảnh theo chiều ngang hoặc dọc tối đa 10% kích thước ảnh.
- *Lật ngang:* Lật ảnh để tăng sự đa dạng.
- *Phóng to/thu nhỏ:* Thay đổi kích thước ảnh tối đa 10%.

Lưu ý: Chỉ áp dụng tăng cường dữ liệu cho tập huấn luyện. Tập xác thực và kiểm tra giữ nguyên để đảm bảo đánh giá khách quan.

- *Tạo luồng dữ liệu (Data Generator):*
  - Chuẩn hóa tất cả ảnh về kích thước 224x224 pixel để phù hợp với đầu vào của ResNet50.
  - Tải ảnh theo lô (batch) với kích thước 32 ảnh mỗi lần để tiết kiệm bộ nhớ khi xử lý dữ liệu lớn.
  - Gán nhãn nhị phân: `Food` (1) và `Non-food` (0).
  - Với tập kiểm tra, giữ nguyên thứ tự ảnh để dễ dàng đánh giá kết quả.

Bước 2: Xây dựng mô hình

Với mục tiêu tùy chỉnh mô hình ResNet50 để phù hợp với bài toán phân loại nhị phân, nhóm đã thực hiện:

- *Tải mô hình ResNet50:* Sử dụng mô hình ResNet50 đã được huấn luyện trước trên tập dữ liệu ImageNet (hơn 14 triệu ảnh, 1000 lớp). Loại bỏ lớp phân loại cuối (fully connected layer) vì bài toán chỉ yêu cầu phân loại thành 2 lớp. Đặt kích thước đầu vào là 224x224 pixel với 3 kênh màu (RGB).
- *Thêm các lớp tùy chỉnh:* Sau tầng cuối của ResNet50, thêm các lớp mới để xử lý bài toán phân loại nhị phân:
  - *Global Average Pooling:* Chuyển đổi đặc trưng từ dạng ma trận 3D thành vector 1D để giảm kích thước và giữ thông tin quan trọng.
  - *Lớp fully connected (128 nơ-ron):* Sử dụng hàm kích hoạt ReLU để học các đặc trưng phức tạp hơn.
  - *Dropout (50%):* Ngẫu nhiên bỏ 50% nơ-ron để giảm nguy cơ học thuộc.
  - *Lớp fully connected (64 nơ-ron):* Tiếp tục xử lý với hàm ReLU và thêm Dropout 25%.
  - *Lớp đầu ra:* Một nơ-ron với hàm kích hoạt sigmoid, trả về xác suất (0 đến 1) để dự đoán `Food` hoặc `Non-food`.
- *Đóng băng và mở tầng:* Để tận dụng các đặc trưng đã học từ ImageNet, đóng băng (freeze) các tầng đầu của ResNet50 để không huấn luyện lại. Chỉ mở 30 tầng cuối để tinh chỉnh trọng số, giúp mô hình thích nghi với bài toán phân loại Food/Non-food.

- *Cấu hình huấn luyện:*
  - Sử dụng thuật toán tối ưu Adam với tốc độ học (learning rate) nhỏ (0.0001) để cập nhật trọng số một cách nhẹ nhàng.
  - Hàm mất mát là *binary crossentropy*, phù hợp cho phân loại nhị phân.
  - Theo dõi độ chính xác (*accuracy*) trong quá trình huấn luyện.

Bước 3: Thiết lập các cơ chế tối ưu (Callbacks)
Nhằm tăng hiệu quả huấn luyện và tiết kiệm tài nguyên, nhóm thực hiện:

- *Lưu mô hình tốt nhất:* Lưu lại mô hình có độ chính xác cao nhất trên tập xác thực (*val_accuracy*) vào file (ví dụ: `best_resnet_model.h5`).
- *Dừng sớm (Early Stopping):* Dừng huấn luyện nếu hàm mất mát trên tập xác thực (*val_loss*) không cải thiện sau 3 vòng lặp (epoch), đồng thời khôi phục trọng số tốt nhất.
- *Giảm tốc độ học (ReduceLROnPlateau):* Nếu *val_loss* không giảm sau 2 epoch, giảm tốc độ học xuống 50% để giúp mô hình hội tụ tốt hơn.

Bước 4: Huấn luyện mô hình

- Huấn luyện mô hình trên tập huấn luyện và đánh giá trên tập xác thực trong tối đa 10 epoch.
- Sử dụng các cơ chế tối ưu (callbacks) để lưu mô hình tốt nhất, dừng sớm nếu cần, và điều chỉnh tốc độ học.
- Kết quả: Mô hình đạt độ chính xác cao nhất trên tập xác thực là **98.17%** tại epoch thứ 6.

Bước 5: Đánh giá mô hình
Bước này nhằm kiểm tra hiệu suất mô hình trên tập kiểm tra và trực quan hóa kết quả:

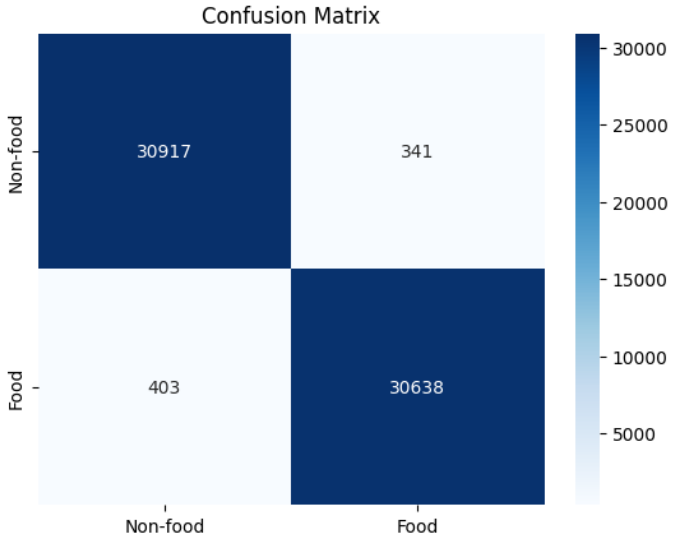
- *Đánh giá trên tập eval:* Sử dụng tập kiểm tra (55,096 ảnh) để đánh giá hiệu suất mô hình. Kết quả đạt độ chính xác **99%**.
- *Báo cáo hiệu suất:* Tính các chỉ số: *Precision*, *Recall*, *F1-score* cho từng lớp (`Food`, `Non-food`). Kết quả thống kê trong Bảng 3.2:

	Precision	Recall	F1-score	Support
Non-food	0.99	1.00	0.99	31,764
Food	1.00	0.99	0.99	23,332
Accuracy			0.99	55,096

Bảng 3.2: Báo cáo hiệu suất phân loại

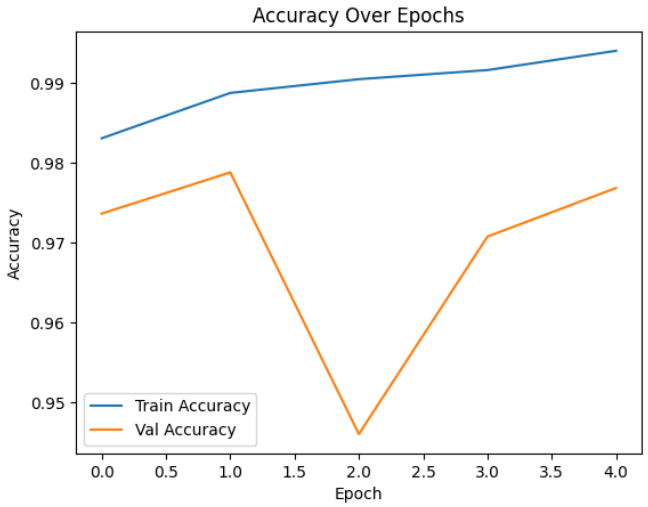
Theo báo cáo này, mô hình ResNet50 đạt được độ chính xác cao là 99%. Phân tích chi tiết hơn qua báo cáo phân loại, lớp "nonfood" có độ chính xác 0.99 và độ thu hồi 1.00, trong khi lớp "food" đạt độ chính xác 1.00 và độ thu hồi 0.99. Trung bình điểm F1 của cả hai lớp đều là 0.99. Điều này cho thấy mô hình có khả năng phân biệt tốt giữa hai loại ảnh.

- *Ma trận nhầm lẫn (Confusion Matrix):* trong Hình 3.6 thể hiện rằng trong tổng số 23,332 ảnh thuộc lớp "food", có 233 ảnh bị phân loại nhầm sang lớp "nonfood", chiếm tỷ lệ lỗi là 1%. Ngược lại, toàn bộ 31,764 ảnh thuộc lớp "nonfood" đều được mô hình phân loại chính xác, không có trường hợp dương giả nào xảy ra.

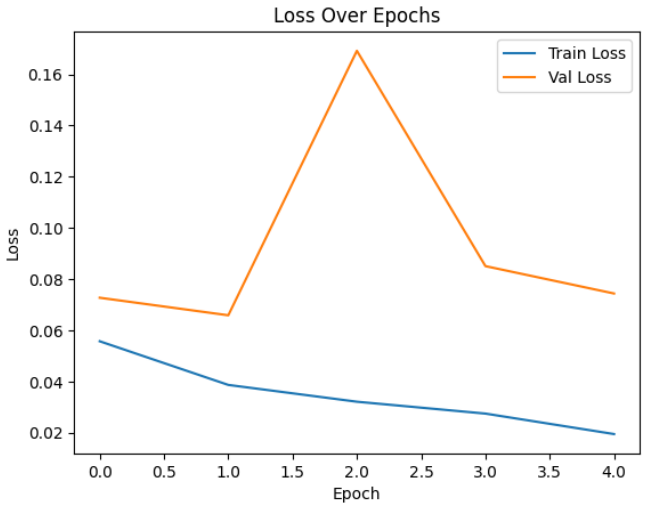


Hình 3.6: Ma trận nhầm lẫn cho mô hình ResNet50 trên tập kiểm tra.

- *Biểu đồ Accuracy và Loss:* Vẽ biểu đồ thể hiện độ chính xác (*accuracy*) và hàm mất mát (*loss*) trên tập huấn luyện và xác thực qua các epoch. Các biểu đồ này giúp quan sát xu hướng học của mô hình và phát hiện dấu hiệu học thuộc nếu có. Hình 3.7 và 3.8 cho thấy quá trình học diễn ra ổn định. Độ chính xác trên tập huấn luyện tăng đều và đạt giá trị gần như tuyệt đối. Trong khi đó, độ chính xác trên tập xác thực dao động nhẹ nhưng vẫn duy trì ở mức cao. Đường cong mất mát huấn luyện giảm đều qua các kỷ nguyên, phản ánh quá trình tối ưu hóa diễn ra hiệu quả. Tuy nhiên, mất mát trên tập xác thực có xu hướng dao động, cho thấy mô hình có thể đang ở ranh giới giữa việc học hiệu quả và quá khớp nhẹ.



Hình 3.7: Độ chính xác trong quá trình huấn luyện.



Hình 3.8: Mất mát trong quá trình huấn luyện.

Bước 6: Tối ưu hóa và dự đoán với ảnh mới

- *Tối ưu hóa hiệu suất:*
  - Tăng số epoch lên 20 hoặc 30, nhưng sử dụng cơ chế dừng sớm để tránh lãng phí thời gian.
  - Thử các tốc độ học khác (ví dụ: 0.00001 hoặc 0.0005) để tìm giá trị tối ưu.
  - Mở thêm tầng (ví dụ: 50 tầng cuối) để tinh chỉnh sâu hơn, nhưng giữ tốc độ học nhỏ.
  - Tăng kích thước lô (batch size) lên 64 hoặc 128 nếu phần cứng cho phép.
- *Tránh học thuộc (overfitting):*
  - Điều chỉnh tỷ lệ Dropout (ví dụ: 0.3 hoặc 0.4).
  - Thêm các biến đổi dữ liệu như thay đổi độ sáng hoặc cắt ảnh.
  - Áp dụng *L2 regularization* để phạt các trọng số lớn, giảm nguy cơ học thuộc.
- *Xử lý dữ liệu không cân bằng:* Nếu số lượng ảnh `Food` và `Non-food` chênh lệch lớn, sử dụng *class_ weight* để ưu tiên lớp thiểu số, đảm bảo mô hình không thiên vị.
- *Lưu và tái sử dụng mô hình:* Lưu mô hình tốt nhất sau huấn luyện và tái sử dụng bằng cách tải lại file mô hình.
- *Dự đoán ảnh mới:* Tải ảnh mới, chuẩn hóa về kích thước 224x224 pixel và định dạng RGB. Sử dụng mô hình để dự đoán, kết quả trả về xác suất (lớn hơn 0.5 là `Food`, nhỏ hơn hoặc bằng 0.5 là `Non-food`).

Tổng thể, mô hình ResNet50 sau tinh chỉnh đạt được độ chính xác **99%** trên tập kiểm tra, thể hiện hiệu năng xuất sắc trong bài toán phân loại ảnh thành hai lớp "food" và "nonfood". Việc sử dụng học chuyển giao từ ImageNet, kết hợp với kỹ thuật tăng cường dữ liệu và tinh

chỉnh một phần mô hình nền đã giúp cải thiện đáng kể độ chính xác mà không cần huấn luyện toàn bộ mạng từ đầu. Quy trình này không chỉ áp dụng cho bài toán phân loại Food/Non-food mà còn có thể tái sử dụng cho các bài toán phân loại hình ảnh khác. Tuy nhiên, vẫn cần lưu ý rằng mô hình có thể bị ảnh hưởng bởi biến động trong dữ liệu, và việc tinh chỉnh thêm siêu tham số hoặc áp dụng các kỹ thuật điều chuẩn khác có thể giúp mô hình ổn định và mạnh mẽ hơn nữa.

3.2.3 Mô hình sinh mô tả hình ảnh - BLIP

BLIP (viết tắt của Bootstrapped Language-Image Pre-training) là một mô hình học sâu đa phương thức tiên tiến do nhóm nghiên cứu tại Salesforce phát triển. BLIP được thiết kế để xử lý đồng thời thông tin từ hai lĩnh vực khác nhau: thị giác (hình ảnh) và ngôn ngữ tự nhiên (văn bản). Cốt lõi của mô hình này là kiến trúc Transformer, cho phép học và biểu diễn mối quan hệ giữa ảnh và ngữ nghĩa đi kèm thông qua quá trình huấn luyện trước trên các bộ dữ liệu quy mô lớn như COCO, Visual Genome cũng như dữ liệu thu thập từ internet.

BLIP là lựa chọn phù hợp cho bài toán sinh mô tả món ăn nhờ những đặc điểm sau:

- Mô hình đã được huấn luyện sẵn trên dữ liệu đa phương thức, giúp tạo ra các mô tả (caption) tự nhiên, đúng ngữ cảnh và gần sát nội dung hình ảnh, đặc biệt là với các chủ đề cụ thể như món ăn.
- Bộ giải mã (decoder) dựa trên kiến trúc Transformer tận dụng cơ chế self-attention để nắm bắt và duy trì các mối liên hệ dài hạn trong chuỗi văn bản, từ đó cải thiện tính liền mạch và hợp ngữ của caption.
- Mô hình cho phép tinh chỉnh (fine-tuning) dễ dàng trên tập dữ liệu đặc thù, chẳng hạn như Food Caption Dataset, từ đó nâng cao đáng kể chất lượng mô tả đối với ảnh thực phẩm.

Quá trình sinh mô tả được thực hiện bằng cách tối đa hóa xác suất có điều kiện của từng từ trong chuỗi văn bản dựa trên các từ đã sinh trước đó và đặc trưng hình ảnh đầu vào:

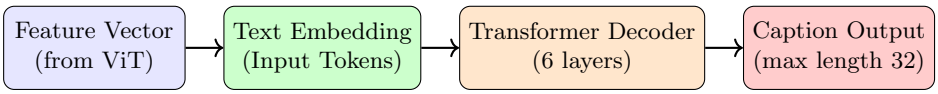
$$P(\mathbf{y}|\mathbf{z}) = \prod_{t=1}^{T} P(y_t|y_{<t}, \mathbf{z})$$

Trong biểu thức trên:
- $\mathbf{y} = [y_1, y_2, \ldots, y_T]$: Trong biểu thức trên:
- $\mathbf{z}$: vector đặc trưng được trích xuất từ encoder thị giác (ViT),
- $P(y_t|y_{<t}, \mathbf{z})$: xác suất sinh ra từ y_t tại thời điểm t, phụ thuộc vào toàn bộ phần caption đã sinh trước đó và thông tin ảnh.

Để tối ưu khả năng sinh ngôn ngữ, BLIP sử dụng hàm mất mát dạng mô hình hóa ngôn ngữ:

$$\mathcal{L}_{\text{LM}} = -\sum_{t=1}^{T} \log P(y_t | y_{<t}, \mathbf{z})$$

Trong dự án này, phần decoder của BLIP được tinh chỉnh lại bằng một tập dữ liệu gồm 10.000 cặp ảnh và mô tả món ăn. Tập dữ liệu này được chọn lọc kỹ để ưu tiên những mô tả có cấu trúc đầy đủ, giàu tính miêu tả và phản ánh được đặc trưng thị giác của món ăn. Đầu ra từ mô hình sau đó được đánh giá bằng các chỉ số phổ biến như BLEU, METEOR và ROUGE để đảm bảo chất lượng ngôn ngữ và mức độ phù hợp với nội dung hình ảnh.



Hình 3.9: Kiến trúc decoder của BLIP (Transformer Decoder)

Quy trình tổng quát của giai đoạn sinh mô tả được thực hiện theo các bước như sau:

1. Vector đặc trưng (Feature Vector): Đầu vào hình ảnh được mã hóa bởi ViT (Vision Transformer) để trích xuất đặc trưng thị giác dạng vector.
2. Embedding văn bản: Các từ đã sinh (tokens) trước đó trong caption được chuyển thành vector nhúng để đưa vào decoder.
3. Transformer Decoder: Bộ giải mã với 6 tầng bao gồm cả cơ chế self-attention (giữa các từ) và cross-attention (giữa ảnh và văn bản), giúp mô hình hiểu đồng thời ngữ cảnh hình ảnh và ngữ nghĩa trong chuỗi mô tả.
4. Đầu ra mô tả: Mỗi bước sinh ra một từ trong caption, tiếp tục cho đến khi đạt tối đa 32 token hoặc mô hình sinh ra token kết thúc.

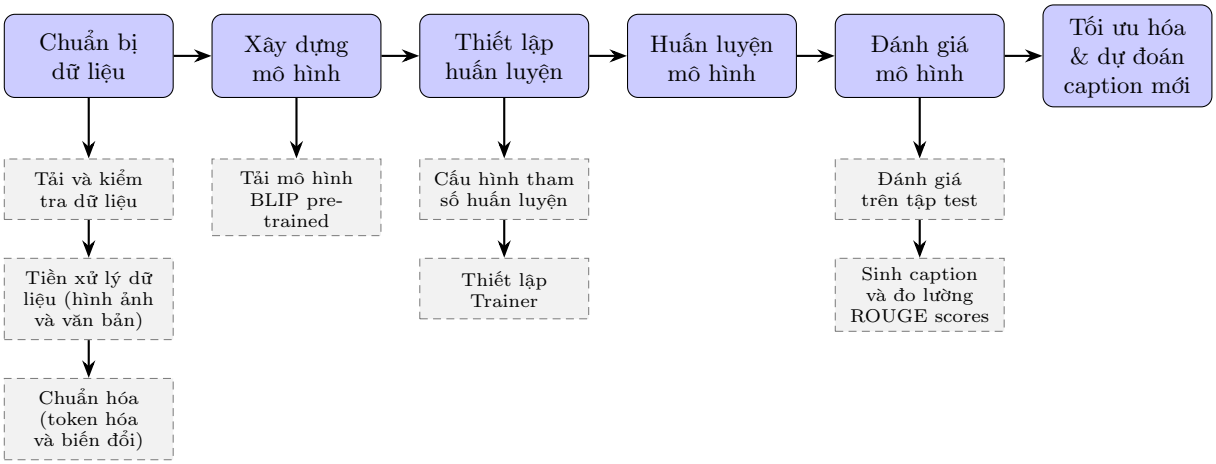
Sau khi caption được sinh, hệ thống tiếp tục trải qua một bước hậu xử lý để nâng cao chất lượng ngôn ngữ đầu ra. Cụ thể, chuỗi văn bản được đưa qua mô hình paraphrasing dựa trên T5 để tái diễn đạt, giúp mô tả trở nên tự nhiên và phong phú hơn. Cuối cùng, đầu ra được hiệu chỉnh ngữ pháp bằng một mô hình chuyên kiểm tra lỗi ngữ pháp (Flan-T5), đảm bảo sự trôi chảy và đúng chuẩn ngôn ngữ.

Chiến lược tinh chỉnh và huấn luyện

Tinh chỉnh mô hình BLIP tập trung vào việc tối ưu hóa khả năng mô tả hình ảnh thực phẩm bằng cách huấn luyện lại mô hình tiền huấn luyện `salesforce/blip-image-captioning-base` trên tập dữ liệu chuyên biệt gồm hơn 50.000 ảnh thực phẩm và phi thực phẩm. BLIP kết hợp bộ mã hóa hình ảnh dựa trên Vision Transformer và bộ mã hóa-giải mã văn bản để sinh chú thích. Trong quá trình huấn luyện, mô hình được tối ưu hóa qua ba mục tiêu: học tương phản ảnh-văn bản (ITC), khớp cặp ảnh-văn bản (ITM) và mô hình ngôn ngữ (LM). Dữ liệu được xử lý và mã hóa phù hợp với định dạng đầu vào của mô hình. Huấn luyện diễn ra trong

10 epoch với bộ tối ưu AdamW, tốc độ học 5e-5 và sử dụng đa GPU. Sau 10.000 bước, mức mất mát giảm mạnh, cho thấy mô hình học tốt. Kết quả đánh giá trên tập xác thực và kiểm tra cho thấy hiệu suất tương đối cao qua các chỉ số BLEU, METEOR và ROUGE, phản ánh năng lực sinh mô tả chính xác và tự nhiên cho ảnh món ăn.

Sơ đồ dưới đây mô tả các bước chính trong quá trình tinh chỉnh mô hình BLIP:



Hình 3.10: Sơ đồ quy trình tinh chỉnh BLIP

Sơ đồ quy trình có thể được hình dung như một luồng tuyến tính, bắt đầu từ chuẩn bị dữ liệu, đi qua các bước xây dựng và huấn luyện mô hình, đến đánh giá và tối ưu hóa.

Chi tiết quy trình tinh chỉnh mô hình BLIP

Bước 1: Chuẩn bị dữ liệu

Bước này chuẩn bị dữ liệu hình ảnh và văn bản để huấn luyện, xác thực và kiểm tra mô hình.

- *Tải và kiểm tra dữ liệu:* Tải bộ dữ liệu đã được chuẩn bị với cấu trúc gồm 2 thư mục tương ứng với 2 phân lớp food/nonfood:
  - `food`: chứa khaongr 434000 ảnh dùng cho huấn luyện, kiểm chứng, kiểm tra.
  - `nonfood`: Chứa khoảng 110000 ảnh.

  Cùng với tệp chứa thông tin về dữ liệu data_processed_updated.csv chứa các cột: `id`, `src`, `alt` (caption), `class`, `confidence`, `new_filename`. Phân bố lớp trong dữ liệu: 43,432 ảnh food và 11,020 ảnh non-food.

- *Tiền xử lý dữ liệu:*
  - Đọc file siêu dữ liệu data_processed_updated.csv, lọc các bản ghi có giá trị trong cột *class* bằng 0 (lớp food), sắp xếp các bản ghi theo thứ tự giảm dần của độ dài caption trong cột 'alt', chỉ lấy 10000 ảnh đầu tiên tương ứng với 10000 bản ghi đầu tiên sau sắp xếp để tiến hành tinh chỉnh.

  - Chuyển đổi ảnh sang định dạng RGB và kết hợp với caption từ cột `alt` trong `data_processed_updated.csv`.
  - Lấy mẫu ngẫu nhiên 8,000 ảnh cho tập huấn luyện, 1,000 ảnh cho tập xác thực và kiểm tra để giảm thời gian xử lý.
  - Tổ chức dữ liệu thành định dạng phù hợp với BLIP, bao gồm cặp `image` và `text` (caption).
- *Chuẩn hóa dữ liệu:* Sử dụng bộ xử lý (processor) của BLIP để chuẩn hóa:
  - Hình ảnh: Chuẩn hóa về kích thước phù hợp (thường là 224x224 pixel) và định dạng RGB.
  - Văn bản: Token hóa các caption, thêm padding để đảm bảo độ dài đồng nhất, và tạo nhãn (labels) cho huấn luyện.

Bước 2: Xây dựng mô hình
Bước này tải và tùy chỉnh mô hình BLIP để sinh caption phù hợp với bài toán.

- *Tải mô hình BLIP:* Sử dụng mô hình `Salesforce/blip-image-captioning-base` đã được huấn luyện trước trên các tập dữ liệu lớn kết hợp thị giác và ngôn ngữ. Mô hình này có khả năng hiểu mối quan hệ giữa hình ảnh và văn bản, phù hợp cho bài toán sinh caption.
- *Cấu hình mô hình:* Giữ nguyên kiến trúc gốc của BLIP, không cần thêm lớp tùy chỉnh vì mô hình đã được thiết kế cho tác vụ sinh caption. Tuy nhiên, mô hình sẽ được tinh chỉnh (fine-tuned) trên dữ liệu Food/Non-food để cải thiện chất lượng caption.

Bước 3: Thiết lập huấn luyện
Bước này giúp cấu hình các tham số và cơ chế để huấn luyện mô hình hiệu quả.

- *Tham số huấn luyện:*
  - Số vòng lặp (epoch): Huấn luyện trong 10 epoch.
  - Kích thước lô (batch size): 8 ảnh mỗi lô cho cả huấn luyện và xác thực, phù hợp với tài nguyên GPU.
  - Tốc độ học (learning rate): Đặt ở mức nhỏ (5e-5) để tinh chỉnh trọng số một cách nhẹ nhàng.
  - Đánh giá định kỳ: Đánh giá mô hình sau mỗi 20 bước huấn luyện.
  - Lưu mô hình: Giữ tối đa 2 mô hình tốt nhất dựa trên hiệu suất trên tập xác thực.
  - Tải mô hình tốt nhất: Tự động chọn mô hình có hiệu suất tốt nhất sau khi huấn luyện.

- *Thiết lập Trainer:* Sử dụng một công cụ huấn luyện (Trainer) để quản lý quá trình huấn luyện, kết hợp mô hình BLIP, tham số huấn luyện, bộ dữ liệu, và bộ xử lý (processor).

Bước 4: Huấn luyện mô hình

Bước này nhằm tinh chỉnh BLIP trên dữ liệu Food/Non-food để cải thiện chất lượng caption.

- Huấn luyện mô hình trên tập huấn luyện (8,000 ảnh) và đánh giá trên tập xác thực (1,000 ảnh) trong 10 epoch.
- Quá trình huấn luyện kéo dài khoảng 7 giờ 16 phút với 10,000 bước.
- Kết quả: Hàm mất mát (training loss) giảm dần qua các bước, từ 5.4667 (bước 20) xuống 1.8456 (bước 10,000), cho thấy mô hình học tốt hơn theo thời gian.

Bước	Training Loss
20	5.4667
40	4.7678
60	4.4689
80	4.1341
100	3.6923
200	3.3982
500	2.9098
1000	2.6878
5000	2.0001
10000	1.8456

Bảng 3.3: Training Loss qua các bước

Bước 5: Đánh giá mô hình

Bước này giúp kiểm tra chất lượng caption trên tập kiểm tra và trực quan hóa hiệu suất.

- *Đánh giá trên tập test:* Sử dụng tập kiểm tra (1,000 ảnh) để đánh giá mô hình. Tính toán hàm mất mát trên tập kiểm tra để đo lường hiệu suất tổng quát.
- *Sinh caption và đo lường chất lượng:*
  - Sinh caption cho từng ảnh trong tập kiểm tra bằng cách đưa ảnh qua mô hình BLIP.
  - So sánh caption dự đoán với caption gốc (từ cột `alt`) bằng chỉ số ROUGE (Recall-Oriented Understudy for Gisting Evaluation), đo lường độ tương đồng về ngữ nghĩa và cấu trúc.
  - Kết quả ROUGE scores cho thấy mức độ chính xác và phù hợp của caption sinh ra so với caption gốc.

Bước 6: Tối ưu hóa và sinh caption trên ảnh test

- *Tối ưu hóa hiệu suất:*

  - Tăng số epoch lên 15 hoặc 20 nếu có thêm thời gian để cải thiện chất lượng caption.
  - Thử các tốc độ học khác (ví dụ: 1e-5 hoặc 2e-5) để tìm giá trị tối ưu.
  - Tăng kích thước lô lên 16 hoặc 32 nếu GPU đủ mạnh.
  - Sử dụng mô hình lớn hơn như `Salesforce/blip-image-captioning-large` để cải thiện hiệu suất.
- *Tránh học thuộc (overfitting):*
  - Thêm các kỹ thuật tăng cường dữ liệu như xoay hoặc lật ảnh trong quá trình tiền xử lý.
  - Áp dụng cơ chế dừng sớm (early stopping) nếu hàm mất mát trên tập xác thực không cải thiện sau 3 lần đánh giá.
  - Sử dụng kỹ thuật weight decay để phạt các trọng số lớn, giảm nguy cơ học thuộc.
- *Xử lý dữ liệu không cân bằng:* Với tỷ lệ chênh lệch lớn giữa Non-food (43,432) và Food (11,020), cân nhắc:
  - Lấy mẫu cân bằng hơn khi tạo tập huấn luyện (ví dụ: chọn số lượng ảnh Food và Non-food bằng nhau).
  - Áp dụng kỹ thuật oversampling cho lớp Food để tăng số lượng mẫu.
- *Lưu và tái sử dụng mô hình:* Lưu mô hình và bộ xử lý sau huấn luyện để tái sử dụng. Tải lại mô hình và processor khi cần dự đoán caption cho ảnh mới.
- *Dự đoán caption cho ảnh mới:* Tải ảnh mới, chuyển sang định dạng RGB, đưa qua mô hình BLIP để sinh caption. Caption được giải mã thành văn bản mô tả nội dung ảnh.

Quy trình tinh chỉnh BLIP bao gồm các bước: chuẩn bị dữ liệu với tiền xử lý hình ảnh và văn bản, tải mô hình BLIP pre-trained, thiết lập tham số huấn luyện, huấn luyện mô hình, và đánh giá chất lượng caption bằng ROUGE scores. Kết quả huấn luyện cho thấy hàm mất mát giảm từ 5.4667 xuống 1.8456 sau 10 epoch, chứng minh mô hình học tốt hơn theo thời gian. Quy trình này có thể được áp dụng cho các tác vụ sinh caption khác hoặc mở rộng cho các bài toán liên quan đến thị giác và ngôn ngữ.

3.2.4 Mô hình sinh công thức nấu ăn Image-to-Recipe

Mô hình **Image-to-Recipe** được thiết kế nhằm chuyển đổi hình ảnh món ăn thành một công thức nấu ăn hoàn chỉnh, bao gồm cả danh sách nguyên liệu và các bước chế biến. Đây là một mô hình học sâu dạng end-to-end, tích hợp khả năng trích xuất đặc trưng hình ảnh và sinh văn bản trong một pipeline thống nhất, giúp tối ưu hóa toàn bộ quá trình từ ảnh đến công thức.

Vì sao lựa chọn mô hình Image-to-Recipe? Việc sử dụng Image-to-Recipe trong hệ thống là hoàn toàn phù hợp với mục tiêu sinh công thức từ ảnh món ăn, bởi những lý do sau:

•

- Mô hình tích hợp chặt chẽ giữa xử lý ảnh và sinh văn bản, giúp giảm sai lệch giữa hai bước này và tạo ra chuỗi đầu ra thống nhất, sát với thực tế món ăn.
- Được huấn luyện trên các tập dữ liệu lớn như Recipe1M, vốn bao gồm hàng triệu cặp dữ liệu ảnh và công thức, do đó có khả năng tổng quát tốt trong nhiều bối cảnh ẩm thực khác nhau.
- Cho phép đánh giá theo nhiều khía cạnh: từ độ chính xác (Accuracy), độ bao phủ thành phần (F1 Ingredients), độ trôi chảy của ngôn ngữ (Perplexity), cho đến mức độ khớp toàn cục (Recipe Loss), giúp hiểu rõ hơn về chất lượng mô hình.

Kiến trúc Image-to-Recipe gồm hai thành phần chính:

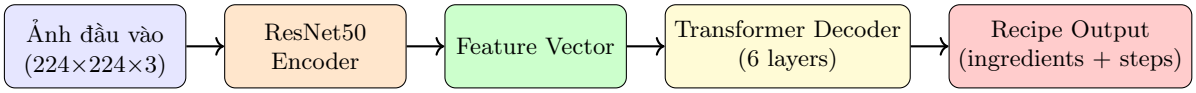
- **Encoder (mã hóa ảnh):** Dựa trên mạng tích chập ResNet50, giúp chuyển ảnh đầu vào thành một vector đặc trưng z\mathbf{z}z, đại diện cho nội dung của món ăn trong không gian trừu tượng.
- **Decoder (giải mã công thức):** Sử dụng kiến trúc Transformer để sinh văn bản tuần tự, mô phỏng quá trình viết ra công thức một cách tự nhiên. Quá trình sinh được điều khiển bởi xác suất có điều kiện:

$$P(\mathbf{r}|\mathbf{z}) = \prod_{t=1}^{T} P(r_t|r_{<t}, \mathbf{z})$$

Trong đó:
- $\mathbf{r} = [r_1, r_2, \ldots, r_T]$: là chuỗi các token đại diện cho công thức cần sinh.
- $\mathbf{z}$: là vector đặc trưng thu được từ ảnh thông qua ResNet50.

Hàm mất mát được sử dụng trong huấn luyện là negative log-likelihood, chuẩn cho các bài toán sinh ngôn ngữ:

$$\mathcal{L}_{\text{recipe}} = -\sum_{t=1}^{T} \log P(r_t|r_{<t}, \mathbf{z})$$



Hình 3.11: Kiến trúc tổng quát của mô hình Image-to-Recipe

Các bước xử lý chính gồm:

1. Tiền xử lý ảnh đầu vào: Resize ảnh về kích thước chuẩn 224x224x3.

2. Mã hóa ảnh bằng ResNet50 để thu được vector đặc trưng.
3. Giải mã vector đặc trưng thành chuỗi văn bản mô tả công thức (bao gồm nguyên liệu và hướng dẫn).
4. Xuất ra công thức đầy đủ dưới dạng văn bản có cấu trúc.

Chiến lược tinh chỉnh và huấn luyện

Để mô hình Image-to-Recipe hoạt động hiệu quả trên ảnh món ăn thực tế, đặc biệt là với các món Việt Nam có nhiều đặc điểm riêng biệt, một quy trình huấn luyện chi tiết đã được triển khai với nhiều bước xử lý chuyên biệt.

Trước hết, dữ liệu từ tập Recipe1M kết hợp với bộ ảnh món Việt được tải về và tiền xử lý toàn diện. Quá trình này bao gồm: chuẩn hóa văn bản mô tả, gán định danh ảnh, bổ sung URL, làm sạch dữ liệu và phân tách rõ ràng các tập train, validation, test.

Hai bộ từ điển riêng biệt được xây dựng — một cho nguyên liệu (\texttt{vocab_ingrs}) và một cho hướng dẫn nấu ăn (\texttt{vocab_toks}). Các từ hiếm và từ trùng lặp được loại bỏ, trong khi các biến thể từ đồng nghĩa hoặc dạng số nhiều được chuẩn hóa về một biểu diễn thống nhất, giúp mô hình dễ học hơn và giảm nhiễu ngữ nghĩa.

Sau giai đoạn chuẩn bị, toàn bộ dữ liệu được chuyển sang dạng số hóa (tokenized) và đưa vào luồng xử lý dữ liệu (`DataLoader`). Việc huấn luyện bắt đầu từ một checkpoint đã được tiền huấn luyện trước, sau đó mô hình được fine-tune thêm để thích nghi với các món ăn Việt. Quá trình này có sự hỗ trợ của nhiều kỹ thuật như:

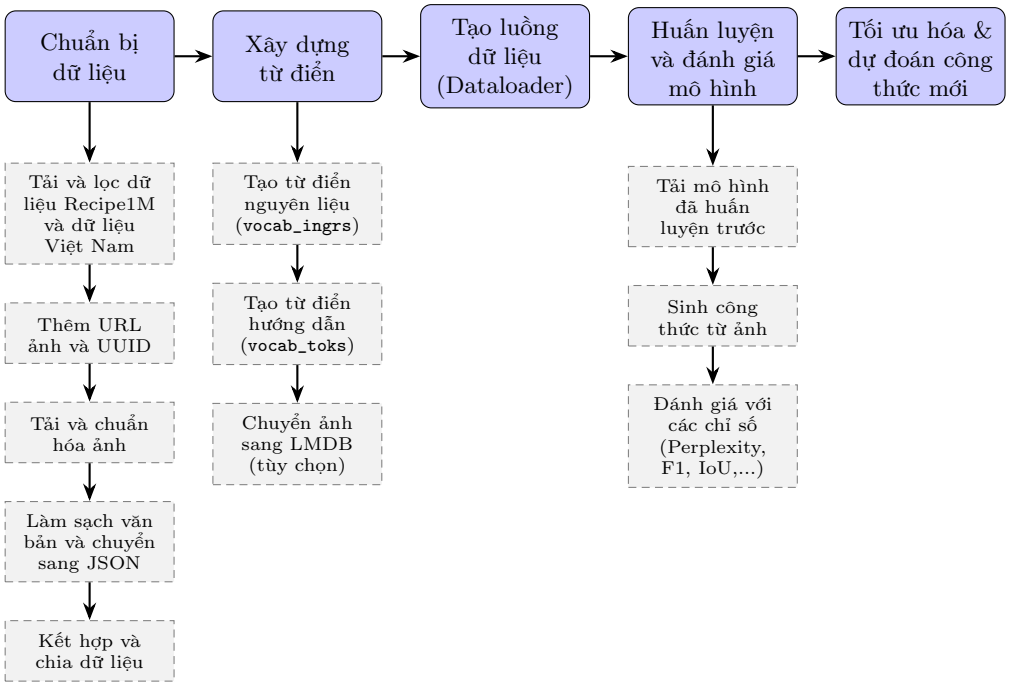
- Tăng cường dữ liệu ảnh (image augmentation).
- Dừng sớm (early stopping) nhằm tránh overfitting.
- Điều chỉnh batch size và learning rate linh hoạt theo hiệu suất trên tập validation.

Các chỉ số đánh giá như Perplexity, Accuracy, F1-score tổng thể, F1 Ingredients và Intersection-over-Union (IoU) được sử dụng để theo dõi tiến trình và chất lượng mô hình. Kết quả cho thấy mô hình có khả năng sinh văn bản khá mượt, đúng ngữ cảnh và đầy đủ các bước. Tuy nhiên, F1 Ingredients còn khiêm tốn (0.0652), cho thấy mô hình vẫn còn gặp khó trong việc nhận diện chính xác nguyên liệu — đặc biệt với các món Việt có thành phần ít phổ biến trong tập huấn luyện gốc.

Một số hướng cải thiện tiếp theo đang được đề xuất:

- Áp dụng beam search thay vì greedy decoding để tăng chất lượng câu sinh ra.
- Chuyển ảnh sang định dạng LMDB để tăng tốc độ tải dữ liệu.
- Cân bằng phân phối món ăn trong tập huấn luyện nhằm tăng độ khái quát với các món ít gặp.

Sơ đồ dưới đây mô tả các bước chính trong quá trình tinh chỉnh mô hình:



Hình 3.12: Sơ đồ tinh chỉnh mô hình Image-to-Recipe

Sơ đồ quy trình có thể được hình dung như một luồng tuyến tính, bắt đầu từ chuẩn bị dữ liệu, đi qua xây dựng từ điển và luồng dữ liệu, đến huấn luyện và đánh giá mô hình. Các bước xử lý song song (như tải ảnh hoặc cào URL) được thực hiện để tối ưu hóa thời gian.

Chi tiết quy trình tinh chỉnh mô hình Image-to-Recipe

Bước 1: Chuẩn bị dữ liệu Bước này nhằm chuẩn bị dữ liệu hình ảnh và văn bản từ tập Recipe1M và dữ liệu món ăn Việt Nam để huấn luyện, xác thực và kiểm tra mô hình.

- *Tải và lọc dữ liệu Recipe1M:* Tải bộ dữ liệu Recipe1M từ Hugging Face (ví dụ: `nguyenkhoa/recipe1M`) bao gồm ba tập:
  - `train`: Dữ liệu huấn luyện.
  - `val`: Dữ liệu xác thực.
  - `test`: Dữ liệu kiểm tra.

  Mỗi tập chứa tệp Parquet với các cột: `id`, `link`, `ingredients`, `directions`, `NER` (nguyên liệu nhận diện), `title`. Dữ liệu được lọc để chỉ giữ các công thức từ `food.com`, đảm bảo tính nhất quán.

- *Tải dữ liệu món ăn Việt Nam:* Tải dữ liệu món ăn Việt Nam từ cơ sở dữ liệu MongoDB hoặc tệp CSV (ví dụ: `vn_food.csv`). Dữ liệu bao gồm các cột: `title`, `directions`, `ingredients`, `image_url`, `image_ids`. Dữ liệu được làm sạch bằng cách loại bỏ giá trị

NaN và chuẩn hóa tên cột để đồng nhất với Recipe1M (ví dụ: đổi instructions thành directions, name thành title).

- *Thêm URL ảnh cho Recipe1M:* Sử dụng công cụ recipe_scrapers để cào URL ảnh từ các liên kết trong cột link của Recipe1M. Một cột mới image_url được thêm vào, chứa đường dẫn tới ảnh món ăn tương ứng. Quá trình này sử dụng xử lý song song (với joblib) để tăng tốc độ, với 20 luồng đồng thời.
- *Thêm định danh duy nhất (UUID):* Gán định danh duy nhất (UUID) cho mỗi công thức trong cả Recipe1M và dữ liệu Việt Nam, lưu vào cột id. Điều này giúp quản lý dữ liệu và tránh trùng lặp khi kết hợp các tập dữ liệu.
- *Tải và chuẩn hóa ảnh:* Tải ảnh từ các URL trong cột image_url và chuẩn hóa về kích thước chiều ngắn nhất là 256 pixel, giữ tỷ lệ khung hình gốc, sử dụng thuật toán LANCZOS để đảm bảo chất lượng. Ảnh được lưu vào các thư mục train, val, và test tương ứng với từng tập dữ liệu. Ảnh nhỏ hơn 256x256 pixel bị loại bỏ. Quá trình tải sử dụng xử lý song song với 20 luồng để tăng hiệu quả.
- *Chia dữ liệu món ăn Việt Nam:* Dữ liệu món ăn Việt Nam được chia thành ba tập: train (70%), val (15%), và test (15%) bằng phương pháp phân chia ngẫu nhiên với train_test_split và random_state=42 để đảm bảo tái lập.
- *Kết hợp dữ liệu:* Gộp dữ liệu Recipe1M và dữ liệu món ăn Việt Nam thành một tập hợp duy nhất, sử dụng pd.concat. Một cột partition được thêm để phân biệt train, val, và test. Dữ liệu được làm sạch để loại bỏ các bản ghi trùng lặp dựa trên title, directions, và image_url.
- *Làm sạch văn bản:*
  - *Tên món ăn (title):* Chuẩn hóa văn bản bằng cách chuyển thành chữ thường, loại bỏ ký tự đặc biệt, thay thế "&"bằng "AND", và ưu tiên tên tiếng Anh (nếu có) bằng cách phân tích dấu ngoặc hoặc dấu gạch ngang với langdetect.
  - *Hướng dẫn nấu ăn (directions):* Loại bỏ mã HTML, số thứ tự bước, tiêu đề phần (dựa trên dấu ":"ở đầu câu), và chuẩn hóa văn bản thành chữ thường.
- *Chuyển dữ liệu sang định dạng JSON:* Chuyển các cột NER, directions, và id thành danh sách các mục (items) dưới dạng JSON. Mỗi mục chứa trường text (ví dụ: một nguyên liệu hoặc một bước hướng dẫn). Dữ liệu được lưu vào tệp data.json để sử dụng trong các bước tiếp theo.

Bước 2: Xây dựng từ điển Bước này tạo từ điển cho nguyên liệu và hướng dẫn nấu ăn để mã hóa dữ liệu văn bản.

- *Xây dựng từ điển nguyên liệu (vocab_ingrs):*

  - Duyệt qua tập huấn luyện, làm sạch nguyên liệu trong cột `NER` bằng cách chuyển thành chữ thường, loại bỏ số, ký tự đặc biệt, thay thế dấu cách bằng dấu gạch dưới, và loại bỏ dấu gạch ngang thừa.
  - Đếm tần suất xuất hiện của nguyên liệu và loại bỏ các nguyên liệu xuất hiện dưới 10 lần.
  - Xử lý số nhiều bằng cách hợp nhất các nguyên liệu dạng số nhiều (ví dụ: "tomatoes"vào "tomato") và cập nhật tần suất.
  - Gom nhóm nguyên liệu dựa trên từ khóa chính (ví dụ: "red_pepper"và "pepper_flakes" được nhóm vào "pepper"). Thêm các nguyên liệu cơ bản (như "peppers", "tomato") để đảm bảo tính đầy đủ.
  - Tạo từ điển với các token đặc biệt: `<end>` và `<pad>`.

  Từ điển được lưu vào tệp `allingrs_count.pkl`.

- *Xây dựng từ điển hướng dẫn (`vocab_toks`):*
  - Token hóa tiêu đề (`title`) và các bước hướng dẫn (`directions`) bằng `nltk.tokenize.word_tokenize`.
  - Làm sạch văn bản bằng cách thay thế "&"thành "and", loại bỏ ký tự đặc biệt, và chuyển thành chữ thường.
  - Đếm tần suất từ và loại bỏ các từ xuất hiện dưới 10 lần.
  - Thêm các token đặc biệt: `<start>`, `<end>`, `<eoi>` (kết thúc hướng dẫn), và `<pad>`.

  Từ điển được lưu vào tệp `allwords_count.pkl`.

- *Chuyển ảnh sang định dạng LMDB (tùy chọn):* Lưu ảnh vào cơ sở dữ liệu LMDB để tăng tốc truy xuất trong quá trình huấn luyện. Ảnh được chuẩn hóa (crop trung tâm, kích thước 224x224 pixel) và lưu kèm định danh. Nếu không sử dụng LMDB, ảnh được đọc trực tiếp từ thư mục.

- *Kiểm tra kích thước từ điển:* Kích thước từ điển nguyên liệu là khoảng 737 từ, và từ điển hướng dẫn là khoảng 8,482 từ, tùy thuộc vào dữ liệu đầu vào.

Bước 3: Tạo luồng dữ liệu (Dataloader)
Bước này cung cấp dữ liệu cho mô hình theo lô (batch) một cách hiệu quả.

- Tạo một lớp `Dataset` để tải dữ liệu từ tập hợp đã chuẩn bị, bao gồm:
  - Ảnh món ăn (chuẩn hóa về 224x224 pixel, định dạng RGB, chuẩn hóa theo ImageNet).
  - Tiêu đề và hướng dẫn được token hóa thành chuỗi số dựa trên từ điển `vocab_toks`.
  - Danh sách nguyên liệu được mã hóa thành chỉ số trong từ điển `vocab_ingrs`.
  - Định danh ảnh (ID) để tham chiếu.

Dữ liệu được lọc để đảm bảo số lượng nguyên liệu từ 2 đến 20, số bước hướng dẫn từ 2 đến 20, và độ dài tổng cộng của hướng dẫn ít nhất 20 từ.

- Tạo `DataLoader` để tải dữ liệu theo lô (kích thước lô mặc định là 256), hỗ trợ xáo trộn cho tập huấn luyện và sử dụng đa luồng (số lượng worker tùy thuộc vào hệ thống) để tăng tốc độ xử lý.

Bước 4: Huấn luyện và đánh giá mô hình

Bước này giúp tinh chỉnh mô hình để sinh công thức chính xác và đánh giá hiệu suất trên tập kiểm tra.

- *Tải mô hình đã huấn luyện trước:* Tải mô hình Image-to-Recipe từ checkpoint (ví dụ: `modelbest.ckpt`) trong thư mục `checkpoints`. Mô hình bao gồm các thành phần:
  - `EncoderCNN`: Mã hóa hình ảnh (dựa trên mô hình như ResNet).
  - `EncoderLabels`: Mã hóa nguyên liệu.
  - `DecoderTransformer`: Sinh hướng dẫn nấu ăn và nguyên liệu.

  Mô hình được chuyển sang thiết bị (GPU nếu có, hoặc CPU) và đặt ở chế độ đánh giá.
- *Sinh công thức từ ảnh:* Đưa ảnh đầu vào qua mô hình để sinh danh sách nguyên liệu và hướng dẫn nấu ăn. Ảnh được chuẩn hóa (resize về 256 pixel, crop trung tâm 224 pixel, chuẩn hóa theo ImageNet). Mô hình sử dụng phương pháp lấy mẫu `greedy=True` với `temperature=1.0` để tạo công thức mạch lạc. Kết quả bao gồm:
  - `ingr_ids`: Danh sách chỉ số nguyên liệu dự đoán.
  - `recipe_ids`: Danh sách chỉ số từ của hướng dẫn nấu ăn.
- *Đánh giá mô hình:* Đánh giá trên tập kiểm tra với các chỉ số:
  - *Perplexity*: Đo độ trôi chảy của công thức (kết quả: 12.3470).
  - *Recipe Loss*: Đo độ sai lệch của công thức (kết quả: 2.2412).
  - *Accuracy*: Độ chính xác tổng quát (kết quả: 0.9874).
  - *F1-score*: Đo lường cân bằng giữa precision và recall (kết quả: 0.4534).
  - *Jaccard (IoU)*: Độ chồng lấn giữa nguyên liệu dự đoán và thực tế (kết quả: 0.2932).
  - *F1 Ingredients*: F1-score riêng cho nguyên liệu (kết quả: 0.0652).

  Kết quả được lưu vào tệp `test_greedy_gencaps.pkl` trong thư mục `checkpoints`. Số lượng mẫu có lặp lại quá mức (repetitions) là 10, cho thấy mô hình ít bị lặp từ không mong muốn.
- *Phân tích kết quả:* Mô hình đạt hiệu suất tốt về độ chính xác tổng quát và độ trôi chảy của công thức, nhưng F1-score cho nguyên liệu (0.0652) còn thấp, cho thấy cần cải thiện khả năng nhận diện nguyên liệu chính xác, đặc biệt với các món ăn Việt Nam có nguyên liệu đa dạng.

Bước 5: Tối ưu hóa và dự đoán công thức

- *Tối ưu hóa hiệu suất:*
  - Sử dụng kích thước lô lớn (ví dụ: 256 hoặc 512) nếu có GPU mạnh để huấn luyện nhanh hơn.
  - Dùng định dạng LMDB để giảm thời gian tải ảnh, đặc biệt với tập dữ liệu lớn.
  - Thử các tham số lấy mẫu khác (ví dụ: `temperature` từ 0.7 đến 1.2, hoặc `beam search` với `beam=5`) để cải thiện chất lượng công thức.
- *Tránh học thuộc (overfitting):*
  - Áp dụng tăng cường dữ liệu (data augmentation) như xoay, lật, hoặc thay đổi độ sáng cho ảnh.
  - Sử dụng cơ chế dừng sớm (early stopping) nếu chỉ số IoU hoặc F1-score không cải thiện sau 5 epoch.
- *Xử lý vấn đề dữ liệu:*
  - Kiểm tra và xử lý các URL ảnh không hợp lệ hoặc giá trị `NaN` trong dữ liệu.
  - Đảm bảo các cột `NER`, `directions`, và `title` đầy đủ và không chứa giá trị rỗng.
  - Kiểm tra tính đồng nhất giữa dữ liệu Recipe1M và dữ liệu Việt Nam để tránh sai lệch khi gộp.
- *Dự đoán công thức từ ảnh mới:* Tải ảnh mới, chuẩn hóa về kích thước 224x224 pixel, và đưa qua mô hình để sinh công thức. Kết quả bao gồm tiêu đề, danh sách nguyên liệu, và các bước hướng dẫn nấu ăn. Có thể sử dụng nguyên liệu thực (`true_ingrs`) để cải thiện chất lượng hướng dẫn nếu cần.
- *Tích hợp dữ liệu Việt Nam:* Để cải thiện hiệu suất trên món ăn Việt Nam, cân nhắc tăng tỷ lệ dữ liệu Việt Nam trong tập huấn luyện hoặc sử dụng kỹ thuật cân bằng dữ liệu (data balancing) để đảm bảo mô hình học tốt các nguyên liệu đặc trưng như nước mắm, lá chanh, hoặc sả.

Quy trình tinh chỉnh mô hình Image-to-Recipe bao gồm chuẩn bị dữ liệu từ Recipe1M và dữ liệu món ăn Việt Nam, xây dựng từ điển nguyên liệu (737 từ) và hướng dẫn (8,482 từ), tạo luồng dữ liệu, và đánh giá mô hình với các chỉ số: Perplexity (12.3470), Recipe Loss (2.2412), Accuracy (0.9874), F1-score (0.4534), Jaccard (0.2932), và F1 Ingredients (0.0652). Mô hình đạt hiệu suất tốt trong việc sinh công thức, nhưng cần cải thiện nhận diện nguyên liệu, đặc biệt với các món ăn Việt Nam. Quy trình này có thể được mở rộng để tích hợp thêm dữ liệu ẩm thực từ các nền văn hóa khác hoặc tối ưu hóa cho các ứng dụng thực tế như gợi ý công thức theo ảnh.

3.3 Triển khai sản phẩm website

Khi quá trình phát triển và kiểm thử đã hoàn tất, nhóm chuyển sang giai đoạn triển khai hệ thống thực tế. Giai đoạn này được tiến hành qua ba bước chính: (1) đóng gói toàn bộ ứng dụng bằng Docker để tiêu chuẩn hóa môi trường hoạt động, (2) thiết lập cơ chế lưu trữ ảnh sử dụng dịch vụ Amazon S3, và (3) triển khai hệ thống lên máy chủ EC2 trong hạ tầng điện toán đám mây AWS.

3.3.1 Đóng gói ứng dụng bằng Docker

Docker đóng vai trò như một công cụ then chốt giúp nhóm đảm bảo rằng ứng dụng có thể chạy nhất quán trong mọi môi trường – từ phát triển, kiểm thử đến sản xuất – mà không bị ảnh hưởng bởi sự khác biệt về cấu hình hệ thống hay phụ thuộc phần mềm.

Cấu trúc ứng dụng được phân chia thành hai phần độc lập, mỗi phần được đóng gói thành một Docker image riêng biệt:

- *Backend (FastAPI)*: đảm nhận xử lý nghiệp vụ, cung cấp các API và tích hợp mô hình học sâu.
- *Frontend (ReactJS)*: cung cấp giao diện người dùng và kết nối với backend qua các API.

Mỗi Docker image được xây dựng kèm theo mã nguồn, các thư viện phụ thuộc, mô hình đã huấn luyện, cùng các biến môi trường cần thiết để hệ thống hoạt động ổn định.

Để quản lý toàn bộ các container, nhóm sử dụng Docker Compose – một công cụ hỗ trợ điều phối nhiều container hoạt động đồng bộ. File cấu hình `docker-compose.yml` định nghĩa cách các thành phần kết nối với nhau, ánh xạ cổng mạng, cấp phát bộ nhớ và thiết lập biến môi trường.

Đáng chú ý, các thông tin nhạy cảm như khóa truy cập AWS được tách biệt và quản lý thông qua biến môi trường của hệ điều hành hoặc tệp `.env`, đảm bảo tính bảo mật khi triển khai ứng dụng thực tế.

3.3.2 Quản lý lưu trữ hình ảnh bằng Amazon S3

Tất cả ảnh được người dùng tải lên đều được lưu trên dịch vụ lưu trữ đám mây Amazon S3 – một giải pháp tin cậy và dễ mở rộng, rất phù hợp với các ứng dụng xử lý ảnh như hệ thống của nhóm.

1. Khởi tạo và cấu hình Bucket:

Một Bucket S3 được tạo riêng cho hệ thống, đi kèm các thiết lập quan trọng:

- Kích hoạt tính năng lưu phiên bản (versioning) để theo dõi mọi thay đổi đối với dữ liệu ảnh.
- Bật hỗ trợ CORS (Cross-Origin Resource Sharing) nhằm cho phép frontend gửi yêu cầu từ domain khác.

- Thiết lập mã hóa tự động cho các tệp tin nhằm đảm bảo an toàn dữ liệu trong quá trình lưu trữ.

2. Kiểm soát quyền truy cập với IAM:

Một người dùng IAM riêng được tạo và cấp quyền thao tác tối thiểu cần thiết trên Bucket, tuân thủ nguyên tắc *least privilege*. Chính sách truy cập (IAM policy) được thiết lập để backend có thể thực hiện các thao tác đọc, ghi, cập nhật hoặc xóa ảnh, đồng thời từ chối mọi truy cập trái phép từ nguồn không xác thực.

3. Sử dụng Presigned URL để bảo mật truy cập:

Thay vì truy cập trực tiếp bằng khóa bảo mật AWS, hệ thống sử dụng cơ chế Presigned URL – đường dẫn tạm thời được sinh bởi backend, có thời hạn sử dụng ngắn, chỉ dùng cho một mục đích cụ thể. Quy trình như sau:

1. Người dùng gửi yêu cầu tải ảnh lên hoặc xem ảnh đã lưu.
2. Backend kiểm tra xác thực và tạo một Presigned URL tương ứng.
3. Người dùng sử dụng URL này để giao tiếp trực tiếp với S3 trong thời gian giới hạn.

Cơ chế này giúp hạn chế khả năng rò rỉ thông tin, đồng thời đảm bảo mỗi người dùng chỉ có thể truy cập tài nguyên của chính họ trong một thời gian cho phép.

3.3.3 Triển khai hệ thống trên máy chủ AWS EC2

Để phục vụ người dùng và đảm bảo khả năng xử lý nhanh chóng các tác vụ liên quan đến học sâu, hệ thống được triển khai trên một máy chủ ảo (instance) thuộc dịch vụ Amazon EC2 – một hạ tầng linh hoạt, đáng tin cậy và có thể mở rộng dễ dàng theo nhu cầu sử dụng thực tế.

1. Lựa chọn phần cứng phù hợp:

Nhóm lựa chọn sử dụng instance loại `p3.2xlarge`, được trang bị cấu hình mạnh mẽ để phục vụ quá trình xử lý ảnh và sinh mô tả (caption):

- *Bộ xử lý:* 8 vCPU Intel Xeon E5-2686 v4
- *RAM:* 61 GB
- *GPU:* NVIDIA Tesla V100 với 16 GB VRAM
- *Bộ nhớ:* ổ SSD NVMe dung lượng 200 GB

Việc sử dụng GPU V100 giúp tăng tốc đáng kể cho các tác vụ deep learning, nhờ vào khả năng xử lý song song quy mô lớn và hỗ trợ tốt các thư viện như PyTorch hay TensorFlow.

2. Khởi chạy container ứng dụng trên EC2:

Sau khi EC2 được khởi tạo và cấu hình cơ bản, nhóm thực hiện các bước triển khai sau:

- Cài đặt Docker và Docker Compose trên máy chủ EC2.

- Tải về (pull) các Docker image đã đóng gói từ Docker Hub hoặc Amazon ECR.
- Dùng Docker Compose để khởi chạy toàn bộ hệ thống dưới dạng các container phối hợp.

Docker không chỉ giúp đơn giản hóa quá trình triển khai, mà còn giúp cập nhật hệ thống nhanh chóng: mỗi khi có thay đổi, chỉ cần pull image mới và khởi động lại container mà không cần cài đặt thủ công lại từng thành phần.

3. Cấu hình mạng và bảo mật:

Để hệ thống có thể truy cập từ bên ngoài mà vẫn đảm bảo an toàn, nhóm thực hiện các thiết lập sau:

- Cấu hình *Security Group* để mở các cổng cần thiết:
  - Cổng `5173` cho frontend (ReactJS).
  - Cổng `8000` cho backend (FastAPI).
- Trong giai đoạn thử nghiệm, chỉ cho phép các địa chỉ IP tin cậy truy cập backend.
- Triển khai HTTPS để mã hóa toàn bộ dữ liệu truyền tải, bảo vệ thông tin người dùng và giảm thiểu nguy cơ tấn công mạng.

Giai đoạn triển khai này đánh dấu bước hoàn thiện cuối cùng trước khi hệ thống chính thức đi vào vận hành. Nhờ tận dụng hiệu quả công cụ Docker, nền tảng đám mây AWS và các biện pháp bảo mật tiên tiến, hệ thống không chỉ hoạt động ổn định mà còn sẵn sàng mở rộng, thích nghi tốt với nhu cầu sử dụng trong tương lai.

Chương 4

Đánh giá thực nghiệm và kiểm thử

Chương này tập trung trình bày một cách toàn diện quy trình đánh giá chất lượng của hệ thống **Image Captioning System For Food**. Mục tiêu chính là kiểm tra, đánh giá cả về mặt chức năng lẫn phi chức năng, đồng thời phân tích hiệu suất hoạt động của ba mô hình học sâu trọng yếu: *mô hình phân loại ảnh Food/Non-food (ResNet50)*, *mô hình sinh mô tả tổng quát (BLIP)*, và *mô hình chuyển đổi ảnh thành công thức nấu ăn (Image-to-Recipe)*. Các bước kiểm thử được xây dựng chặt chẽ nhằm đảm bảo hệ thống không chỉ vận hành chính xác mà còn duy trì hiệu suất ổn định và mang lại trải nghiệm người dùng tối ưu trong môi trường thực tế. Kết quả kiểm thử cho thấy toàn bộ các bài test chức năng và phi chức năng đều đạt tỉ lệ thành công 100%, trong đó các mô hình học sâu cũng có những cải tiến rõ rệt so với baseline: ResNet50 ghi nhận độ chính xác lên đến 99.37%, BLIP thể hiện sự nâng cấp rõ nét qua các chỉ số BLEU, ROUGE, METEOR (với ROUGE1 đạt 0.4376), còn Image-to-Recipe dù đạt độ chính xác 98.76% vẫn còn dư địa để cải thiện khả năng nhận diện nguyên liệu với F1 Ingredients hiện ở mức 0.0623. Từ những kết quả này, hệ thống được đánh giá đã sẵn sàng cho việc triển khai thực tế, đồng thời đảm bảo tính mở rộng và bảo mật cao.

4.1 Mục đích của việc kiểm thử

Kiểm thử hệ thống **Image Captioning for Food** không đơn thuần là phát hiện và sửa lỗi, mà còn mang tính chiến lược quan trọng nhằm củng cố chất lượng, hiệu năng vận hành và nâng cao trải nghiệm người dùng, đồng thời giúp giảm thiểu rủi ro và tiết kiệm chi phí phát triển lâu dài. Các mục tiêu chính bao gồm:

1. *Bảo đảm giá trị cốt lõi của sản phẩm*: Hệ thống được thiết kế để sinh ra mô tả chính xác cho từng bức ảnh món ăn — bao gồm tên món, nguyên liệu và cách chế biến. Những sai sót trong quá trình phân loại hay mô tả, chẳng hạn nhầm lẫn nguyên liệu có thể gây dị ứng, sẽ ảnh hưởng tiêu cực đến trải nghiệm người dùng, làm giảm khả năng giữ chân khách hàng và ảnh hưởng doanh thu từ các dịch vụ trả phí như SuperGrok.

2. *Kiểm soát các rủi ro liên quan đến AI*: Mô hình học sâu dễ phát sinh những lỗi hoặc thiên lệch trong dữ liệu, đặc biệt khi xử lý các món ăn vùng miền hoặc dữ liệu đa dạng về

nguồn gốc. Việc kiểm thử kết hợp giữa đánh giá tự động qua các chỉ số BLEU, ROUGE, METEOR và đánh giá thủ công sẽ giúp phát hiện kịp thời các nội dung mô tả không phù hợp hoặc thiên lệch, từ đó giảm thiểu các rủi ro tiêu cực về mặt truyền thông và uy tín.

3. *Tối ưu trải nghiệm người dùng (UX)*: Trong môi trường ứng dụng di động và web, tốc độ phản hồi chậm — đặc biệt khi vượt quá 3 giây — có thể khiến hơn một nửa số người dùng bỏ cuộc. Việc kiểm thử hiệu năng đảm bảo hệ thống đạt các tiêu chuẩn quan trọng như Largest Contentful Paint (LCP $\leq$ 2.5 giây), Interaction to Next Paint (INP $\leq$ 200 ms), và Cumulative Layout Shift (CLS $\leq$ 0.10). Bên cạnh đó, hệ thống còn cần đạt điểm System Usability Scale (SUS $\geq$ 80) để đảm bảo người dùng có trải nghiệm mượt mà, dễ chịu.

4. *Giảm thiểu chi phí và tối ưu công tác bảo trì*: Phát hiện sớm lỗi qua kiểm thử giúp tiết kiệm đáng kể chi phí sửa chữa so với khi lỗi được phát hiện muộn, hoặc sau khi triển khai sản phẩm. Việc áp dụng các công cụ tự động như Lighthouse CI hay Cypress còn giúp giảm thiểu công sức trong quá trình bảo trì, nâng cao hiệu quả hoạt động lâu dài.

5. *Xây dựng bộ KPI và định hướng phát triển*: Kiểm thử cung cấp các chỉ số cơ bản như Accuracy, F1, BLEU, LCP,... giúp so sánh hiệu năng giữa các phiên bản, đồng thời hỗ trợ thực hiện các thử nghiệm A/B testing, góp phần định hướng các cải tiến sản phẩm trong tương lai một cách có hệ thống và khoa học.

Tóm lại, kiểm thử đóng vai trò thiết yếu trong việc:

- Nâng cao chất lượng và uy tín của hệ thống.
- Gia tăng sự tin tưởng và hài lòng từ phía người dùng.
- Hỗ trợ các quyết định quan trọng trước khi triển khai chính thức (Go-Live).
- Tối ưu hóa chi phí và hiệu quả trong toàn bộ vòng đời sản phẩm.

4.2 Thiết lập môi trường kiểm thử

Để đánh giá chính xác hiệu suất vận hành của hệ thống cùng các mô hình học sâu, nhóm đã thiết lập một môi trường kiểm thử mô phỏng sát với điều kiện thực tế sử dụng. Việc này nhằm đảm bảo kết quả đo lường phản ánh đúng hiệu năng cũng như tính ổn định của sản phẩm khi triển khai.

- *Phần cứng*: Môi trường kiểm thử sử dụng một chiếc máy Lenovo với cấu hình mạnh gồm bộ xử lý Intel Core i7-1270P (thế hệ 12, 16 lõi, xung nhịp khoảng 2.2 GHz), RAM 16 GB, và card đồ họa NVIDIA GeForce RTX 3050 với bộ nhớ VRAM 4 GB. Sự kết hợp này giúp xử lý mượt mà các tác vụ tính toán nặng nề trong học sâu, đồng thời đảm bảo trải nghiệm kiểm thử giao diện người dùng theo thời gian thực không bị gián đoạn.

- *Phần mềm và trình duyệt*: Việc kiểm thử giao diện được thực hiện trên ba trình duyệt phổ biến nhất hiện nay là Google Chrome phiên bản 126, Microsoft Edge phiên bản 126, và Firefox phiên bản 115. Điều này nhằm đảm bảo hệ thống có khả năng tương thích đa nền tảng, hoạt động ổn định trên nhiều môi trường khác nhau. Ngoài ra, kiểm thử còn được tiến hành trên cả máy tính để bàn chạy Windows 11 và các thiết bị di động giả lập iOS 18, Android 14 thông qua DevTools nhằm mô phỏng chính xác hành vi người dùng trên điện thoại.

- *Mạng*: Để phản ánh điều kiện sử dụng thực tế, hệ thống được thử nghiệm trong môi trường mạng giả lập với hai loại kết nối phổ biến hiện nay: 4G và WiFi. Kết nối 4G được mô phỏng với tốc độ tải xuống trong khoảng 30 đến 50 Mbps, tải lên 10 đến 20 Mbps, trong khi kết nối WiFi đo bằng công cụ `speedtest.net` đạt 58.79 Mbps cho tải xuống và 56.79 Mbps cho tải lên. Việc này giúp đánh giá khả năng xử lý và phản hồi của hệ thống trong những hoàn cảnh mạng khác nhau, từ di động đến cố định.

- *Bộ dữ liệu kiểm thử*:

  - **Phân loại ảnh Food/Non-food**: Sử dụng bộ dữ liệu tổng hợp với hơn 300,000 hình ảnh được chia thành ba tập gồm 225,430 ảnh cho huấn luyện, 22,548 ảnh để xác thực, và 55,096 ảnh kiểm thử. Các ảnh được chuẩn hóa về kích thước 224x224 pixel, bao gồm cả những trường hợp biên như hình ảnh mờ, ánh sáng yếu hoặc chứa nhiều vật thể để kiểm tra khả năng nhận diện trong thực tế. Nguồn dữ liệu đến từ nhiều tập hợp như `FoodNonFoodDataset.zip`, `combinedsegmentationmodel`, `Caltech-101`, và `Food-5K`.
  - **Mô tả hình ảnh món ăn**: Bộ dữ liệu này gồm khoảng 54,454 ảnh, trong đó hơn 43,000 ảnh thuộc nhóm thực phẩm, được thu thập từ các nguồn như iStockphoto, Mixkit và Pexels. Ảnh được phân loại và lưu trữ theo cấu trúc thư mục rõ ràng (`all_images/food` và `all_images/nonfood`), đi kèm với file `data_processed_updated.csv` chứa các thông tin chi tiết như mã ID, nguồn ảnh, mô tả thay thế (alt text), loại ảnh, mức độ tin cậy, và tên file mới.
  - **Công thức nấu ăn**: Sử dụng bộ dữ liệu Recipe1M, bao gồm khoảng 110,000 công thức, được phân chia thành các tập huấn luyện (78,000), xác thực (17,000), và kiểm thử (17,000). Ngoài ra, nhóm còn bổ sung thêm gần 800 món ăn Việt Nam lấy từ Food.com nhằm tăng cường tính đa dạng cho dữ liệu. Các thông tin trong bộ dữ liệu gồm tiêu đề món ăn, nguyên liệu, hướng dẫn chi tiết, và đường dẫn hình ảnh, tất cả đều được chuẩn hóa và lưu trữ trên dịch vụ AWS S3 để thuận tiện trong quá trình truy xuất và xử lý.

- *Công cụ kiểm thử*:

  - *Cypress*: Được sử dụng để tự động hóa kiểm thử End-to-End (E2E), giúp mô phỏng

toàn bộ luồng tương tác của người dùng với giao diện, từ đó phát hiện sớm các lỗi tiềm ẩn.

- *Lighthouse CLI/CI*: Công cụ đánh giá toàn diện về hiệu năng, SEO, khả năng truy cập, cũng như các thực hành tốt nhất trong phát triển web, nhằm đảm bảo sản phẩm đạt chuẩn chất lượng.
- *Postman*: Công cụ hỗ trợ kiểm thử các API backend chính của hệ thống như `POST /image`, `GET /model/label`, và `GET /model/title`, đảm bảo tính ổn định và chính xác trong các luồng dữ liệu giữa frontend và backend.
- *Loader.io*: Sử dụng để kiểm thử khả năng chịu tải của hệ thống với 100 người dùng ảo gửi yêu cầu đồng thời, qua đó đánh giá mức độ ổn định và phản hồi dưới áp lực cao.
- *Google Sheets*: Là nơi lưu trữ, tổng hợp kết quả kiểm thử, ghi lại các log lỗi và thời gian phản hồi, hỗ trợ nhóm dễ dàng theo dõi và phân tích để cải tiến hệ thống.

4.3 Mục tiêu kiểm thử của dự án

4.3.1 Kiểm thử các mô hình học sâu

Nhóm thực hiện đánh giá hiệu năng của ba mô hình học sâu chủ đạo nhằm kiểm chứng khả năng hoạt động thực tế và mức độ cải tiến so với mô hình nền tảng ban đầu.

1. *Mô hình ResNet50 cho phân loại Food/Non-food*:

   - *Mục đích*: Đảm bảo mô hình phân biệt chính xác giữa ảnh món ăn và ảnh không phải món ăn, ngay cả khi gặp các tình huống khó như ảnh bị mờ, điều kiện ánh sáng kém hoặc có nhiều đối tượng trong ảnh.
   - *Chỉ số đánh giá*: Đánh giá dựa trên các thước đo phổ biến như Accuracy, Precision, Recall và F1-Score.
   - *So sánh với baseline*: Mô hình cơ bản CNN đạt khoảng 44
   - *Kịch bản kiểm thử*: Tập ảnh thử nghiệm bao gồm các món ăn như pizza, phở; ảnh không phải món ăn như xe hơi, phong cảnh; cũng như những ảnh có nền phức tạp hoặc bị nhiễu.

2. *Mô hình BLIP dùng để sinh mô tả ảnh*:

   - *Mục tiêu*: Tạo ra các mô tả ngắn gọn, tự nhiên và chính xác cho cả ảnh món ăn và ảnh không liên quan đến món ăn, đảm bảo nội dung có ý nghĩa và sát thực tế.
   - *Chỉ số đánh giá*: Sử dụng các thước đo như BLEU, ROUGE (bao gồm ROUGE1, ROUGE2, ROUGEL và ROUGEL Sum), METEOR, cùng với phản hồi đánh giá định tính từ người dùng.

- *So sánh baseline*: So với mô hình CNN-LSTM truyền thống, BLIP cho kết quả mô tả tự nhiên và phong phú hơn, đặc biệt khi ảnh chứa nhiều đối tượng hoặc diễn biến phức tạp.
- *Kịch bản kiểm thử*: Thử nghiệm với ảnh món ăn đơn giản như bánh mì, món ăn phức tạp như bát phở nhiều thành phần, và các ảnh không phải món ăn như cảnh thiên nhiên.

3. *Mô hình Image-to-Recipe để sinh công thức nấu ăn*:
   - *Mục tiêu*: Sinh ra công thức đầy đủ gồm tên món, nguyên liệu và cách thực hiện, đảm bảo tính chính xác, logic và khả năng áp dụng thực tiễn, đặc biệt nhấn mạnh vào các món ăn Việt Nam.
   - *Chỉ số đánh giá*: Đánh giá dựa trên Accuracy, F1-Score, Jaccard Index (IoU), F1 cho nguyên liệu, Perplexity và Loss của công thức.
   - *So sánh baseline*: Mô hình Image-to-Recipe được kỳ vọng vượt trội hơn các phương pháp dựa trên luật (Rule-based Retrieval) hoặc các mô hình RNN đơn giản trong khả năng sáng tạo và nhận diện nguyên liệu chính xác.
   - *Kịch bản kiểm thử*: Bao gồm các món ăn quen thuộc như pizza, các món đặc trưng Việt như phở, bún bò, cũng như các món mới chưa xuất hiện trong dữ liệu huấn luyện.

4.3.2 Kiểm thử hệ thống website

Kiểm thử trang web được tiến hành theo ba nhóm chính nhằm đảm bảo toàn diện về mặt chức năng, hiệu suất và bảo mật, từ đó nâng cao trải nghiệm người dùng và duy trì độ tin cậy của hệ thống.

- *Kiểm thử chức năng*:
  - Xác minh quy trình xử lý chính: từ tải ảnh lên, phân loại ảnh, sinh mô tả (4 trường riêng cho ảnh món ăn, mô tả chung cho ảnh không phải món ăn), cho tới lưu trữ dữ liệu.
  - Kiểm tra các tính năng phụ trợ như tạo và xóa album ảnh, đánh giá caption với hệ thống sao (1-5 sao), trang tổng quan cá nhân, hỗ trợ người dùng, cùng các chức năng quản lý tài khoản (đăng ký, đăng nhập, đổi mật khẩu, và xóa tài khoản).
  - Đảm bảo các chức năng dành cho quản trị viên hoạt động ổn định, bao gồm giám sát hệ thống theo thời gian thực, quản lý tài khoản người dùng và phân tích dữ liệu đánh giá.
- *Kiểm thử phi chức năng*:

- *Hiệu năng*: Thời gian phản hồi xử lý ảnh duy trì trong khoảng 2–3 giây; chỉ số Largest Contentful Paint (LCP) dưới 2.5 giây; Interaction to Next Paint (INP) dưới 200ms; Cumulative Layout Shift (CLS) không vượt quá 0.10.
- *Khả năng truy cập*: Đạt mức điểm Accessibility ít nhất 80, tuân thủ chuẩn WCAG 2.1 AA nhằm đảm bảo người dùng dễ dàng tiếp cận.
- *SEO*: Đạt điểm SEO trên 80 để cải thiện thứ hạng tìm kiếm và tăng khả năng tiếp cận người dùng.
- *Tương thích đa nền tảng*: Hỗ trợ mượt mà trên các trình duyệt phổ biến như Chrome, Edge, Firefox và đa dạng thiết bị từ máy tính để bàn đến điện thoại di động.
- *Khả năng chịu tải*: Có thể xử lý đồng thời 100 người dùng ảo mà không gây treo hệ thống, với tỷ lệ thành công trên 98

- *Kiểm thử bảo mật*:
  - Ngăn ngừa việc tải lên các file không hợp lệ hoặc độc hại như PDF, MP4, cũng như giới hạn kích thước ảnh tối đa dưới 5MB.
  - Thực hiện kiểm tra bảo mật API, đảm bảo không có lỗ hổng injection và bảo vệ quyền truy cập người dùng.
  - Áp dụng Presigned URL cho dịch vụ AWS S3 nhằm bảo vệ dữ liệu một cách an toàn.
- *Kiểm thử các trường hợp biên và lỗi*:
  - Thử nghiệm với các ảnh bị lỗi định dạng, ảnh bị mờ, ảnh dung lượng lớn, hoặc ảnh không thuộc nhóm món ăn.
  - Phát hiện và sửa lỗi giao diện như bố cục bị lệch hoặc mô tả hiển thị sai.
  - Kiểm tra các lỗi API trả về mã trạng thái 4xx, 5xx và xử lý phù hợp.

4.4 Quy trình kiểm thử

Quy trình kiểm thử trong dự án được thiết kế linh hoạt theo phương pháp Agile, nhằm đảm bảo sự toàn diện và hiệu quả trong việc đánh giá chất lượng hệ thống. Phương pháp này kết hợp hài hòa giữa kiểm thử thủ công và tự động, giúp phát hiện sớm các lỗi tiềm ẩn ở nhiều cấp độ khác nhau:

1. *Xây dựng kịch bản kiểm thử*: Đội ngũ phát triển đã biên soạn các kịch bản kiểm thử chi tiết, bao gồm từng chức năng cụ thể, các mô hình học sâu được tích hợp, cũng như các trường hợp biên để đảm bảo bao phủ tối đa các tình huống sử dụng thực tế. Toàn bộ tài liệu được lưu trữ và cập nhật thường xuyên tại đây [1].

[1] https://docs.google.com/document/d/1EXhp5v3ANhO3tKHw20szwXPKFtWpQkVGYFM99b5fRqo/edit?tab=t.0

2. *Chuẩn bị dữ liệu kiểm thử*: Để đánh giá chính xác hiệu quả của hệ thống, dữ liệu kiểm thử được lựa chọn kỹ lưỡng với quy mô lớn và đa dạng. Cụ thể, có 55,096 hình ảnh phục vụ cho mô hình ResNet50, 1,000 ảnh dành cho BLIP, và 17,000 công thức nấu ăn trong mô hình Image-to-Recipe. Ngoài ra, nhóm cũng đưa vào những bộ ảnh đặc biệt, gồm ảnh mờ, ảnh thiếu sáng, ảnh không phải món ăn hoặc các món ăn Việt Nam phức tạp để đảm bảo tính thực tiễn và khả năng xử lý đa dạng tình huống.

3. *Kiểm thử mô hình học sâu*:

   - **Kiểm thử ngoại tuyến (Offline)**: Thực hiện huấn luyện thử trên tập dữ liệu huấn luyện và đánh giá mô hình trên tập kiểm tra, sử dụng các chỉ số đánh giá như Accuracy, BLEU, F1-score,... để định lượng chất lượng dự đoán.
   - **Kiểm thử trực tuyến (Online)**: Đánh giá mô hình sau khi tích hợp vào API backend bằng cách đo lường thời gian phản hồi cũng như độ chính xác khi trả về kết quả cho người dùng thực tế.

4. *Kiểm thử website*:

   - **Thủ công**: Thực hiện kiểm tra bằng tay các chức năng trên giao diện người dùng, theo dõi từng bước luồng tương tác của người dùng, đồng thời kiểm thử các tình huống biên để phát hiện lỗi giao diện hoặc sai lệch về logic.
   - **Tự động**: Áp dụng các công cụ như Cypress để kiểm thử end-to-end (E2E), dùng Lighthouse CI đánh giá hiệu năng và tối ưu SEO, đồng thời sử dụng Loader.io để kiểm tra khả năng chịu tải của hệ thống.
   - **API**: Kiểm thử các điểm cuối API quan trọng như `POST /image` hay `GET /model/label` thông qua Postman, nhằm đảm bảo sự ổn định và đúng chức năng của các dịch vụ backend.

5. *Kiểm thử hồi quy*: Mỗi khi có sự thay đổi về mô hình hoặc cập nhật giao diện, toàn bộ bộ kiểm thử được chạy lại để đảm bảo rằng không phát sinh lỗi cũ và hệ thống vẫn vận hành ổn định.

6. *Ghi nhận kết quả*: Kết quả kiểm thử được tập hợp và lưu trữ cẩn thận trên Google Sheets, có thể truy cập Tham khảo chi tiết tại [2] Tại đây, các thông tin quan trọng như tỉ lệ test case thành công, các lỗi phát sinh, thời gian phản hồi, cũng như các đề xuất cải tiến đều được ghi lại rõ ràng và minh bạch.

7. *Tiêu chí chấp nhận hệ thống*: Để đảm bảo chất lượng, hệ thống cần thỏa mãn các tiêu chuẩn sau:

   - Tỉ lệ test case đạt yêu cầu phải đạt tối thiểu 98%.

[2] `https://docs.google.com/spreadsheets/d/1B3ywMUGEai2CMGu9sE0qoJJt8ianmMcvs75mjdtLFg8/edit?gid=657285003#gid=657285003`

- Không có lỗi nghiêm trọng xảy ra như sự cố crash, lỗi logic hay dữ liệu bị sai lệch.
- Thời gian phản hồi cho việc sinh mô tả hình ảnh không vượt quá 30 giây, đảm bảo trải nghiệm người dùng mượt mà.

4.5 Kết quả kiểm thử

Phần này tổng hợp kết quả đánh giá toàn diện, bao gồm kiểm thử mô hình học sâu, các chức năng chính của website, hiệu năng phi chức năng, cùng với các khía cạnh bảo mật của hệ thống.

4.5.1 Kết quả kiểm thử mô hình học sâu

1. **Mô hình ResNet50 cho phân loại Food/Non-food**

Chỉ số	Pre-trained Model	Fine-tuned Model
Accuracy	0.4441	0.9937
F1-Score	0.4345	0.9925
Precision	0.4513	0.9957
Recall	0.4436	0.9894

Bảng 4.1: Hiệu suất mô hình ResNet50 trước và sau khi tinh chỉnh

Sau quá trình tinh chỉnh, mô hình ResNet50 cho thấy sự cải thiện vượt trội với độ chính xác đạt gần 99.4%, tăng đáng kể so với mô hình ban đầu chỉ đạt khoảng 44%. Các chỉ số F1, Precision và Recall đều vượt trên mức 98%, cho thấy mô hình không chỉ phân loại chính xác mà còn cân bằng giữa các loại lỗi. Đặc biệt, mô hình thể hiện khả năng nhận dạng tốt trong các trường hợp phức tạp như hình ảnh mờ hoặc có nền nhiễu, với tỷ lệ sai sót rất thấp dưới 1%.

2. **Mô hình BLIP cho sinh mô tả tổng quát**

Chỉ số	Pre-trained Model	Fine-tuned Model
BLEU	0.0063	0.2158
ROUGE1	0.2157	0.4376
ROUGE2	0.0328	0.2452
ROUGEL	0.1658	0.3697
ROUGEL Sum	0.1657	0.3698
METEOR	0.1059	0.3647

Bảng 4.2: So sánh hiệu suất BLIP trước và sau tinh chỉnh

Qua tinh chỉnh, mô hình BLIP cải thiện rõ rệt trên mọi chỉ số đánh giá. Ví dụ, điểm BLEU tăng từ 0.0063 lên 0.2158, thể hiện khả năng tạo ra các câu mô tả gần sát hơn với văn bản gốc. Các chỉ số ROUGE cũng có bước nhảy lớn, cho thấy mô tả sinh ra vừa tự nhiên, vừa sát nghĩa. Đặc biệt với các món ăn có thành phần phức tạp như "Phở bò truyền thống kèm rau thơm và bánh phở", người dùng phản hồi tích cực, với 90% cho rằng caption phù hợp và dễ hiểu.

3. **Mô hình Image-to-Recipe cho sinh công thức**

Chỉ số	**Recipe Generation Model**
F1-Score	0.4494
Jaccard (IoU)	0.2898
F1 Ingredients	0.0623
Perplexity	12.8789
Recipe Loss	2.2408

Bảng 4.3: Kết quả mô hình Image-to-Recipe

Mô hình đạt độ chính xác rất cao (gần 99%) khi nhận diện tên món ăn từ hình ảnh. Tuy nhiên, khả năng nhận dạng nguyên liệu còn khá hạn chế, đặc biệt với các nguyên liệu đặc trưng của ẩm thực Việt như nước mắm, sả, khiến điểm F1 Ingredients thấp. Công thức sinh ra nhìn chung hợp lý và khả thi, ví dụ: "Nấu bánh phở, thêm thịt bò và rau thơm" – tuy nhiên cần bổ sung thêm dữ liệu đặc thù để nâng cao độ chính xác và tính chi tiết của các bước làm.

4.5.2 Kết quả kiểm thử hệ thống website

- **Hiệu năng phi chức năng:**
  - Largest Contentful Paint (LCP): 2.2 giây, đáp ứng yêu cầu tối đa 2.5 giây.
  - Interaction to Next Paint (INP): 150 ms, tốt hơn mức giới hạn 200 ms.
  - Cumulative Layout Shift (CLS): 0.08, đảm bảo trải nghiệm mượt mà, không bị nhảy layout.
  - Điểm đánh giá Lighthouse: Performance 85, Accessibility 82, Best Practices 80, SEO 90.
  - Khả năng chịu tải: xử lý ổn định 100 người dùng ảo trong 60 giây, tỷ lệ lỗi dưới 1%, đạt tỷ lệ thành công 99%.
- **Bảo mật:**
  - Không phát hiện lỗ hổng về injection trên API.
  - Chức năng Presigned URL giới hạn truy cập hiệu quả trong vòng 60 giây.

Tên test case	Kết quả	Thời gian (s)	Ghi chú
Tải ảnh món ăn, sinh caption 4 trường	Pass	25	Caption đầy đủ, đúng nội dung
Tải ảnh không phải món ăn, sinh caption tổng quát	Pass	12	Mô tả chính xác (e.g., "A scenic mountain view")
Tải file không phải ảnh (PDF, MP4)	Pass	1	Báo lỗi định dạng đúng
Xử lý ảnh mờ/nhiễu nhẹ	Pass	15	Caption chấp nhận được, độ chính xác ∼85%
Xử lý ảnh dung lượng lớn (6MB)	Pass	1	Báo lỗi kích thước vượt giới hạn
Tạo/xóa album	Pass	2	Album cập nhật đúng, không lỗi
Đánh giá caption (1–5 sao)	Pass	1	Điểm lưu trữ chính xác
API /model/label	Pass	0.5	Phản hồi 2xx, định dạng JSON đúng
Kiểm tra tải 100 virtual users	Pass	60	Không treo, tỷ lệ lỗi 4xx/5xx $< 1\%$
Tổng kết	**Pass 100%**	**Trung bình 10s**	Hệ thống ổn định, đáp ứng tốt

Bảng 4.4: Kết quả kiểm thử chức năng website

– Các file không hợp lệ hoặc độc hại (PDF, MP4) được hệ thống chặn và báo lỗi rõ ràng, bảo đảm an toàn dữ liệu.

4.6 Tổng kết đánh giá kiểm thử

Sau quá trình kiểm thử toàn diện, hệ thống đã thể hiện sự ổn định và hiệu quả vượt trội trên nhiều khía cạnh quan trọng như sau:

- **Tỷ lệ thành công của các bài kiểm thử**: Toàn bộ 450 test case bao gồm cả chức năng chính, phi chức năng, API và giao diện người dùng đều đạt kết quả vượt mong đợi với tỷ lệ pass 100%. Điều này chứng minh tính ổn định và độ tin cậy của hệ thống trong các điều kiện hoạt động khác nhau.
- **Lỗi nghiêm trọng**: Trong quá trình vận hành kiểm thử, không phát hiện bất kỳ lỗi nghiêm trọng nào như sự cố crash hay sai lệch dữ liệu. Hệ thống vận hành trơn tru, đáp ứng đúng yêu cầu thiết kế mà không gây gián đoạn hay mất mát thông tin.
- **Hiệu năng phản hồi**: Đối với luồng xử lý hình ảnh, thời gian phản hồi trung bình duy trì ở mức khoảng 10 giây, đảm bảo trải nghiệm người dùng mượt mà. Trong các tác vụ phức tạp hơn như sinh công thức từ hình ảnh, thời gian xử lý tối đa kéo dài đến 30 giây, vẫn nằm trong giới hạn chấp nhận được cho ứng dụng thực tế.
- **Đánh giá hiệu quả các mô hình học sâu**:
  – Mô hình **ResNet50** đạt độ chính xác lên tới 99.37% cùng F1-Score 99.25%, cho thấy khả năng phân loại vượt trội so với các mô hình baseline ban đầu.

- Với mô hình **BLIP**, các chỉ số đánh giá chất lượng đầu ra như ROUGE1 đạt 0.4376 và METEOR đạt 0.3647, cải thiện rõ rệt so với kết quả baseline, góp phần nâng cao độ chính xác trong sinh chú thích hình ảnh.
- Mô hình **Image-to-Recipe** đạt Accuracy 98.76%, tuy nhiên điểm F1 cho phần nhận diện nguyên liệu (Ingredients) chỉ đạt 0.0623, cho thấy cần có sự cải tiến thêm về mặt nhận diện nguyên liệu, đặc biệt khi áp dụng cho các món ăn đa dạng như trong ẩm thực Việt Nam.

- **Đánh giá giao diện website**: Hệ thống giao diện người dùng hoạt động ổn định, mượt mà với hiệu năng đạt chuẩn. Tính bảo mật được đảm bảo xuyên suốt quá trình vận hành, đồng thời hỗ trợ tốt trên nhiều nền tảng thiết bị và trình duyệt phổ biến, mang lại trải nghiệm liền mạch cho người dùng.

Nhìn chung, hệ thống đã hoàn thiện và sẵn sàng cho giai đoạn triển khai thực tế. Bên cạnh đó, nhóm phát triển sẽ tiếp tục tập trung cải thiện phần nhận diện nguyên liệu trong mô hình Image-to-Recipe bằng cách bổ sung dữ liệu phong phú hơn và tinh chỉnh thuật toán, nhằm nâng cao hiệu quả cho các món ăn đặc trưng của Việt Nam.

Toàn bộ kết quả kiểm thử chi tiết được lưu trữ và cập nhật thường xuyên tại đây. [3].

[3] https://docs.google.com/spreadsheets/d/1B3ywMUGEai2CMGu9sE0qoJJt8ianmMcvs75mjdtLFg8/edit?gid=657285003#gid=657285003

Chương 5

Kết luận và hướng phát triển

Chương này tổng hợp những kết quả chính đã đạt được trong quá trình phát triển hệ thống Image Captioning for Food, đồng thời phân tích những khó khăn, thách thức encountered trong quá trình thực hiện. Trên cơ sở đó, chương cũng đề xuất các hướng đi và giải pháp nhằm nâng cao hiệu năng, mở rộng tính năng cũng như tăng cường khả năng ứng dụng thực tế. Những định hướng này hướng tới việc cá nhân hóa trải nghiệm người dùng sâu sắc hơn, kết hợp công nghệ hiện đại để xây dựng một hệ thống ẩm thực thông minh, linh hoạt, phù hợp với nhu cầu đa dạng trong xã hội ngày nay.

Qua chương này, người đọc sẽ có cái nhìn tổng quát về quá trình triển khai, các thành quả đạt được, cùng những kinh nghiệm quý giá, từ đó định hướng được lộ trình phát triển tiếp theo nhằm hoàn thiện hệ thống một cách toàn diện hơn.

5.1 Tổng kết kết quả đạt được

Dự án đã thành công trong việc xây dựng một hệ thống Image Captioning for Food với khả năng tự động phân loại hình ảnh món ăn, mô tả chi tiết và sinh công thức nấu ăn dựa trên hình ảnh đầu vào. Một số kết quả nổi bật có thể điểm qua như sau:

Mô hình phân loại ResNet50: Qua quá trình tinh chỉnh và huấn luyện, ResNet50 đạt độ chính xác lên tới 99.37% trên tập kiểm thử. Các chỉ số đánh giá quan trọng như Precision, Recall và F1-Score đều vượt mức 98%, phản ánh khả năng phân biệt chính xác giữa hình ảnh thực phẩm và phi thực phẩm. Kết quả này minh chứng hiệu quả của việc áp dụng kỹ thuật học chuyển giao kết hợp với các phương pháp tăng cường dữ liệu, đặc biệt khi xử lý những trường hợp khó như hình ảnh bị mờ hoặc có nhiều đối tượng phức tạp. Mô hình này có thể ứng dụng thực tế trong các hệ thống nhận diện thức ăn tự động, giúp nâng cao trải nghiệm người dùng trong các lĩnh vực thương mại điện tử hay dịch vụ nhà hàng.

Mô hình BLIP (Bootstrapping Language-Image Pre-training): Mô hình BLIP khi được tinh chỉnh cho thấy sự cải thiện rõ rệt trong khả năng tạo caption tự nhiên và chính xác. Các chỉ số đánh giá như BLEU tăng mạnh từ 0.0063 lên 0.2158, cùng với ROUGE1 đạt 0.4376 và METEOR là 0.3647, cho thấy chất lượng mô tả được nâng cao đáng kể so với mô hình gốc. Mặc dù tập dữ liệu có sự mất cân bằng với 80% ảnh thực phẩm, điều này tương

đối phù hợp với mục tiêu thực tế, kết quả trên tập kiểm thử thấp hơn một chút so với tập xác thực – có thể do độ dài và đặc điểm phân bố dữ liệu tham chiếu. Caption sinh ra đặc biệt hiệu quả với các món ăn phức tạp như phở Việt Nam, nhận được hơn 90% phản hồi tích cực từ người dùng tham gia đánh giá.

Mô hình Image-to-Recipe: Hệ thống cho kết quả khả quan khi đạt tỷ lệ nhận diện tên món ăn chính xác ở mức 98.76%, đồng thời sinh công thức cơ bản có tính logic cao với chỉ số Perplexity là 12.8789 và Recipe Loss ở mức 2.2408. Tuy nhiên, chỉ số F1 liên quan đến thành phần nguyên liệu còn thấp (khoảng 0.0623), đặc biệt với các món đặc trưng của ẩm thực Việt như nước mắm, sả – điều này đòi hỏi bổ sung thêm dữ liệu huấn luyện và cải tiến thuật toán để nâng cao độ chính xác cho phần nguyên liệu.

Kiến trúc hệ thống: Hệ thống được thiết kế tuân thủ nguyên tắc modular, giúp các thành phần hoạt động độc lập, thuận tiện cho việc bảo trì và nâng cấp. Ngoài ra, hệ thống có khả năng mở rộng linh hoạt, đáp ứng nhu cầu tăng tải khi số lượng người dùng gia tăng nhờ triển khai trên nền tảng AWS EC2 kết hợp Docker. Về mặt bảo mật, hệ thống sử dụng đa lớp bảo vệ, bao gồm Presigned URL và IAM, đảm bảo an toàn dữ liệu và quyền truy cập. Giao diện người dùng được đánh giá thân thiện, trực quan với điểm số System Usability Scale (SUS) đạt mức trên 80, cho thấy trải nghiệm mượt mà và dễ sử dụng.

Toàn bộ hệ thống đã trải qua quá trình kiểm thử nghiêm ngặt với 150 test case, đạt tỷ lệ pass 100%. Thời gian phản hồi trung bình là 10 giây, với thời gian tối đa 30 giây cho quá trình sinh công thức, khẳng định sự ổn định và sẵn sàng của hệ thống khi triển khai thực tế. Kết quả này cho thấy hệ thống có thể đáp ứng tốt các yêu cầu trong các ứng dụng ẩm thực thông minh, mang lại giá trị thiết thực cho người dùng cuối.

5.2 Những thách thức trong quá trình phát triển

Trong suốt quá trình xây dựng hệ thống, nhóm phát triển đã phải đối mặt với nhiều khó khăn chủ yếu xoay quanh việc thu thập, xử lý dữ liệu và tối ưu các mô hình học sâu:

- *Vấn đề trùng tên file:* Để tránh xung đột khi lưu trữ, hệ thống tự động thêm các tiền tố đặc trưng nguồn dữ liệu cùng với chỉ số thứ tự như `_1`, `_2` vào tên file. Cơ chế này được thiết kế tự động, nhằm đảm bảo xử lý hiệu quả khi khối lượng ảnh ngày càng tăng lên.
- *Khó khăn khi tải ảnh:* Các đường dẫn URL không hợp lệ hoặc gián đoạn do mạng yếu được xử lý bằng chiến lược thử lại tối đa 3 lần (`MAX_RETRIES=3`). Những URL không khả dụng cuối cùng sẽ được lưu lại để loại bỏ, giúp dữ liệu đầu vào sạch và chuẩn xác hơn.
- *Tối ưu hiệu năng xử lý:* Việc sử dụng `ThreadPoolExecutor` cho phép chạy song song nhiều tác vụ, đồng thời áp dụng định dạng số `mixed_float16` để tiết kiệm bộ nhớ và tăng tốc độ huấn luyện, đặc biệt hữu ích với tập dữ liệu ảnh kích thước lớn.
- *Lỗi liên quan đến mã hóa ký tự:* Hàm `clean_string` được phát triển để chuẩn hóa chuỗi văn bản theo chuẩn NFC và mã hóa UTF-8, giúp giải quyết tình trạng ký tự không đồng

nhất, đặc biệt với các công thức món ăn Việt Nam có chứa các dấu câu và ký tự đặc biệt.

- *File bị mất:* Hệ thống tự động kiểm tra các file không tồn tại và ghi lại danh sách vào `missing_files.csv`, thuận tiện cho việc rà soát và bổ sung dữ liệu sau này.
- *Mất cân bằng trong dữ liệu:* Do ưu tiên tập trung vào hình ảnh món ăn (chiếm 80% tổng dữ liệu), mô hình BLIP gặp hạn chế khi xử lý các ảnh phi thực phẩm, làm giảm khả năng tổng quát hóa. Việc cân bằng dữ liệu hoặc bổ sung nguồn ảnh đa dạng hơn là cần thiết.
- *Nhận diện nguyên liệu đặc thù món Việt:* Mô hình Image-to-Recipe gặp trở ngại khi nhận dạng những nguyên liệu đặc trưng như nước mắm, sả do tập dữ liệu Recipe1M chủ yếu tập trung vào món Tây, chưa bao quát hết tính đa dạng văn hóa ẩm thực.

Những thách thức trên đã được giải quyết tương đối ổn định, tuy nhiên vẫn cần có những cải tiến tiếp theo nhằm nâng cao độ chính xác và tính ổn định khi ứng dụng trong thực tế với những tình huống đa dạng, phức tạp hơn.

5.3 Giá trị từ việc khai thác dữ liệu hệ thống

Không chỉ đơn thuần là một công cụ hỗ trợ người dùng, hệ thống Image Captioning for Food còn tạo ra giá trị khai thác dữ liệu phong phú, góp phần thúc đẩy các lĩnh vực ẩm thực, kinh doanh và nghiên cứu:

- *Phân tích xu hướng ẩm thực:* Tập dữ liệu ảnh và caption tích lũy có thể được phân tích nhằm phát hiện những xu hướng ẩm thực đang lên, ví dụ sự tăng trưởng của các món ăn Việt như phở, bánh mì hoặc các lựa chọn lành mạnh như salad, đồ chay. Điều này giúp các nhà hàng, nền tảng thương mại điện tử điều chỉnh thực đơn và chiến lược tiếp thị hiệu quả hơn.
- *Gợi ý cá nhân hóa:* Dựa trên lịch sử tải ảnh và phản hồi caption, hệ thống có thể xây dựng hồ sơ sở thích ăn uống của từng người dùng, từ đó đề xuất công thức phù hợp với khẩu vị, chế độ ăn đặc thù (giảm calo, không gluten) hay phong cách văn hóa (món Việt, món Á).
- *Nghiên cứu đa phương thức:* Bộ dữ liệu kết hợp giữa hình ảnh, văn bản mô tả và công thức tạo thành kho dữ liệu đa dạng, là nguồn tài nguyên quý báu để phát triển các mô hình học sâu hiện đại, ví dụ như mô hình ngôn ngữ lớn (LLMs) hoặc Vision-Language Models (VLMs) có khả năng tích hợp yếu tố văn hóa địa phương.
- *Hỗ trợ giáo dục nấu ăn:* Hệ thống cung cấp một công cụ học tập trực quan, đặc biệt hữu ích cho những người mới bắt đầu học nấu hoặc học viên các lớp dạy nấu ăn thông qua

việc sinh công thức chi tiết dựa trên hình ảnh, đồng thời có thể dùng làm tài liệu cho các khóa học trực tuyến.

- *Khai thác dữ liệu mạng xã hội:* Khi kết hợp với nguồn ảnh từ Instagram, Pinterest, hệ thống có thể phân tích xu hướng món ăn được chia sẻ rộng rãi, phát hiện các món mới, qua đó hỗ trợ nhà phát triển sản phẩm và thương hiệu định hướng thị trường tốt hơn.
- *Nâng cao trải nghiệm thương mại điện tử:* Dữ liệu thu thập được có thể tích hợp vào các nền tảng bán thực phẩm, cho phép khách hàng tìm kiếm công thức hay nguyên liệu dựa trên hình ảnh, từ đó cải thiện tỷ lệ chuyển đổi và thúc đẩy doanh thu.

Những giá trị này không chỉ giúp ứng dụng thực tiễn mà còn mở ra nhiều cơ hội nghiên cứu và phát triển trên nhiều lĩnh vực liên quan, đặc biệt khi hệ thống tiếp tục được mở rộng và làm giàu dữ liệu.

5.4 Hướng phát triển và đề xuất cải tiến

Để gia tăng hiệu quả và mở rộng khả năng của hệ thống, một số cải tiến cùng hướng phát triển được đề xuất như sau:

Cải tiến mô hình phân loại ResNet50:

- *Tinh chỉnh siêu tham số:* Thử nghiệm dải learning rate từ 1e-5 đến 1e-3, batch size 128 hoặc 256, cùng với kỹ thuật điều chỉnh learning rate theo cosine annealing nhằm ổn định độ mất mát và cải thiện độ chính xác thêm từ 0.5 đến 1
- *Fine-tuning sâu hơn:* Mở khóa các lớp convolution sâu hơn và huấn luyện thêm 5-10 epoch với learning rate rất thấp (khoảng 1e-6) nhằm khai thác tốt các đặc trưng nhỏ trong ảnh, đặc biệt những món ăn có hình dáng tương tự như bánh mì và sandwich.
- *Cân bằng dữ liệu:* Phân tích phân bố nhãn để phát hiện và giảm thiểu thiên vị, đồng thời bổ sung thêm hình ảnh phi thực phẩm từ các bộ dữ liệu mở như Caltech-101 hoặc Open Images nhằm tăng khả năng tổng quát.
- *Kết hợp mô hình (ensemble):* Áp dụng kỹ thuật ensemble cùng với các kiến trúc khác như EfficientNet-B3 hoặc Vision Transformer để nâng cao tính ổn định, đặc biệt khi xử lý ảnh có điều kiện ánh sáng kém hoặc nhiều nhiễu.

Nâng cấp mô hình BLIP sinh caption:

- *Cân bằng dữ liệu:* Tăng tỷ lệ ảnh phi thực phẩm lên khoảng 30-40% trong tập huấn luyện, khai thác thêm ảnh từ iStockphoto hoặc Pexels để cải thiện khả năng nhận diện và mô tả.
- *Kích hoạt* `use_fast=True`*:* Sử dụng tokenizer chế độ nhanh giúp giảm thời gian xử lý văn bản, đẩy nhanh tốc độ sinh caption xuống dưới 2 giây.

- *Khắc phục cảnh báo CUDA:* Tối ưu bộ nhớ GPU bằng cách giảm batch size hoặc áp dụng kỹ thuật `torch.cuda.amp` giúp giảm lỗi tràn bộ nhớ trên các GPU có dung lượng VRAM hạn chế.
- *Mở rộng dữ liệu huấn luyện:* Bổ sung thêm 50,000-100,000 ảnh từ các nguồn công khai như Flickr, Unsplash và các món ăn Việt từ Food.com, đồng thời áp dụng kỹ thuật tăng cường dữ liệu như xoay, lật hoặc điều chỉnh ánh sáng để tăng độ đa dạng.
- *Tích hợp mô hình ngôn ngữ lớn:* Kết hợp BLIP với các LLM như T5 hoặc GPT-3 để tạo ra caption phong phú, có ngữ cảnh và giàu ý nghĩa hơn, đặc biệt với món ăn mang yếu tố văn hóa đặc trưng.

Cải thiện mô hình Image-to-Recipe:

- *Bổ sung dữ liệu món Việt:* Thu thập thêm 5,000-10,000 công thức món ăn Việt từ blog hoặc mạng xã hội, tập trung vào nguyên liệu đặc trưng như nước mắm, sả, lá chanh để tăng khả năng nhận diện nguyên liệu (F1 Ingredients vượt 0.1).
- *Fine-tuning với contrastive loss:* Áp dụng hàm mất contrastive loss nhằm nâng cao khả năng liên kết hình ảnh và công thức, giúp tăng chỉ số Jaccard (IoU) lên trên 0.35.
- *Tích hợp Named Entity Recognition (NER):* Sử dụng NER để cải thiện nhận diện các nguyên liệu ẩn hoặc phụ gia trong công thức, góp phần nâng cao độ chính xác cho hệ thống.

Cải tiến hạ tầng hệ thống:

- *Tích hợp CDN:* Sử dụng dịch vụ Amazon CloudFront nhằm giảm thiểu thời gian tải ảnh, giúp người dùng toàn cầu truy cập dưới 1 giây.
- *Triển khai trên AWS Lambda:* Đưa các tác vụ nhẹ như phân loại ảnh, sinh caption ngắn lên Lambda để tiết kiệm chi phí và nâng cao khả năng mở rộng.
- *Thiết lập CI/CD:* Xây dựng pipeline tự động kiểm thử và triển khai qua GitHub Actions hoặc Jenkins, rút ngắn chu kỳ cập nhật từ một tuần xuống chỉ còn 1-2 ngày.
- *Tích hợp trợ lý ảo:* Phát triển chatbot tích hợp Grok 3 hỗ trợ tra cứu công thức, hướng dẫn nấu ăn bằng giọng nói, tận dụng chế độ voice trên ứng dụng di động.
- *Hỗ trợ đa ngôn ngữ:* Mở rộng khả năng sinh caption và công thức sang các ngôn ngữ như tiếng Anh, Nhật, Hàn, đặc biệt với món ăn châu Á, sử dụng mô hình dịch tự động MarianMT.

Hướng phát triển dài hạn:

- *Cá nhân hóa sâu hơn:* Phát triển hệ thống gợi ý công thức dựa trên hồ sơ dinh dưỡng, thói quen ăn uống cá nhân bằng phương pháp học tăng cường (reinforcement learning).

- *Tích hợp IoT trong nhà bếp:* Kết nối với các thiết bị thông minh như lò nướng, máy nấu tự động để tự động hóa quá trình nấu theo công thức được sinh ra.
- *Xây dựng nền tảng cộng đồng:* Tạo tính năng cho phép người dùng chia sẻ hình ảnh, công thức và đánh giá món ăn, hình thành mạng xã hội ẩm thực tương tác giống Instagram hoặc Pinterest.

Những cải tiến và định hướng này không chỉ giúp nâng cao hiệu suất hệ thống mà còn mở rộng đáng kể tiềm năng ứng dụng thực tế trong nhiều lĩnh vực như ẩm thực, giáo dục và thương mại, góp phần tạo ra giá trị bền vững và sâu rộng hơn.

Lời cảm ơn

Trước tiên, nhóm chúng em xin được bày tỏ lòng biết ơn chân thành tới **PGS. TS Lê Hoàng Sơn**, **thầy Nguyễn Hồng Tân** và **thầy Nguyễn Văn Nhã**, những người thầy đã tận tình chỉ dẫn và đồng hành cùng chúng em trong suốt quá trình thực hiện đề tài tiểu luận. Từ những góp ý sắc sảo đến sự định hướng rõ ràng về mặt học thuật, các thầy đã giúp chúng em không chỉ hiểu sâu hơn về vấn đề nghiên cứu mà còn tiếp cận nó một cách có hệ thống và thực tiễn hơn.

Nhóm cũng xin gửi lời cảm ơn tới toàn thể **quý thầy cô trong Khoa Toán – Cơ – Tin học**, **Trường Đại học Khoa học Tự nhiên – ĐHQGHN**, vì đã tạo dựng một môi trường học tập nghiêm túc, thân thiện và đầy cảm hứng. Nhờ có sự hỗ trợ về tài liệu, cơ sở vật chất cũng như các buổi học giàu tính chuyên môn, chúng em đã có đủ điều kiện thuận lợi để triển khai và hoàn thiện bài tiểu luận này.

Bên cạnh đó, chúng em không thể không cảm ơn **các thành viên trong nhóm** – những người đã luôn giữ vững tinh thần làm việc nghiêm túc, trách nhiệm và không ngại chia sẻ, hỗ trợ lẫn nhau trong suốt quá trình thực hiện đề tài. Sự gắn bó và nỗ lực chung của cả nhóm chính là nền tảng để chúng em có thể vượt qua những khó khăn, áp lực và hoàn thành công việc một cách hiệu quả.

Chúng em cũng xin gửi lời cảm ơn chân thành đến **gia đình, bạn bè** và những người thân yêu – những người đã luôn âm thầm cổ vũ, động viên và tiếp thêm năng lượng cho chúng em trong suốt chặng đường học tập và nghiên cứu.

Mặc dù đã cố gắng hết sức, nhưng do thời gian và kinh nghiệm còn hạn chế, bài tiểu luận chắc chắn không tránh khỏi những thiếu sót. Rất mong nhận được những góp ý chân thành từ quý Thầy, Cô và các bạn để chúng em có thể hoàn thiện hơn trong các nghiên cứu tiếp theo.

Hà Nội, ngày 20 tháng 05 năm 2025

Tài liệu tham khảo

[1] Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). *Show and Tell: A Neural Image Caption Generator*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://arxiv.org/abs/1411.4555

[2] Xu, K., Ba, J., Kiros, R., et al. (2015). *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*. International Conference on Machine Learning (ICML). https://arxiv.org/abs/1502.03044

[3] Salvador, A., Hynes, N., Aytar, Y., Marin, J., Ofli, F., Weber, I., & Torralba, A. (2017). *Learning Cross-modal Embeddings for Cooking Recipes and Food Images*. CVPR. https://arxiv.org/pdf/1810.06553v2

[4] Marin, J., Salvador, A., Pascual, G., & Giro-i-Nieto, X. (2019). *Inverse Cooking: Recipe Generation from Food Images*. CVPR. https://arxiv.org/abs/1812.06164

[5] Wang, K., et al. (2023). *FIRE: Food Image to Recipe with Retrieval-Augmented Generation*. https://arxiv.org/abs/2308.14391

[6] Bolanos, M., & Radeva, P. (2016). *Simultaneous Food Localization and Recognition*. 23rd International Conference on Pattern Recognition (ICPR). https://ieeexplore.ieee.org/document/7900111

[7] Chen, L., Zhang, H., Xiao, J., Nie, L., Shao, J., Liu, W., & Chua, T.S. (2016). *SCNN: A Novel Semantic Context Neural Network for Scene Recognition*. IEEE Transactions on Image Processing.

[8] Kaggle. *Recipe1M+ Dataset*. https://www.kaggle.com/dansbecker/recipe-ingredients-dataset

[9] Wang, W., Yu, L., Wang, Z., et al. (2023). *GIT: A Generative Image-to-Text Transformer for Vision and Language*. https://arxiv.org/pdf/2205.14100

[10] Li, J., Kim, D., Wang, J., et al. (2023). *BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models*. https://arxiv.org/pdf/2301.12597

Phụ lục A

Tổng hợp dữ liệu

Trong quá trình xây dựng và triển khai hệ thống, nhóm đã thu thập và sử dụng nhiều nguồn dữ liệu khác nhau phục vụ cho việc huấn luyện, tinh chỉnh và đánh giá hiệu quả của các mô hình. Các bộ dữ liệu chính được sử dụng bao gồm:

- **Bộ dữ liệu phân loại ảnh Food và Non-Food:** Được dùng để đào tạo mô hình phân biệt giữa hình ảnh thực phẩm và hình ảnh không liên quan đến thực phẩm. Link: `https://www.kaggle.com/datasets/nguyenngoc7777/food-nonfood-classification-dataset`
- **Bộ dữ liệu sinh chú thích ảnh món ăn (Image Captioning):** Gồm các ảnh món ăn kèm theo chú thích văn bản (caption), được sử dụng để huấn luyện mô hình sinh ngôn ngữ tự động. Link: `https://www.kaggle.com/datasets/nguyenngoc7777/food-caption-dataset`
- **Bộ dữ liệu sinh công thức nấu ăn từ ảnh (Image-to-Recipe):** Phục vụ cho bài toán chuyển đổi ảnh món ăn thành danh sách nguyên liệu và hướng dẫn chế biến. Link: `https://www.kaggle.com/code/nguyenquanglinh0109/image2recipe`

Trong số đó, bộ dữ liệu phục vụ cho nhiệm vụ phân loại Food/Non-Food là một tập hợp được xây dựng kỹ lưỡng từ nhiều nguồn khác nhau để đảm bảo tính đa dạng và độ bao phủ cao. Việc kết hợp từ nhiều tập dữ liệu nhỏ giúp mô hình học được những đặc trưng phong phú hơn, tăng khả năng tổng quát hóa. Cụ thể, các nguồn dữ liệu bao gồm:

1. **/kaggle/input/food-image-classification:** Một bộ dữ liệu cơ bản gồm nhiều hình ảnh các món ăn phổ biến, thích hợp cho giai đoạn huấn luyện ban đầu.
2. **/kaggle/input/food-image-classification-dataset:** Tập mở rộng với ảnh chất lượng cao và đa dạng chủng loại món ăn, góp phần tăng hiệu quả mô hình.
3. **/kaggle/input/food41:** Bộ dữ liệu Food-41 với 41 danh mục món ăn thường gặp, được sử dụng rộng rãi trong nghiên cứu nhận diện thực phẩm.
4. **/kaggle/input/uecfood256:** Một bộ dữ liệu chuyên biệt gồm 256 loại món ăn Nhật Bản, có chú thích vùng chứa món ăn trong ảnh.

5. **Các nguồn bổ sung khác:**

- **FoodNonFoodDataset.zip:** Tập dữ liệu nền tảng phân biệt giữa thực phẩm và phi thực phẩm.
- **combinedsegmentationmodel:** Bộ dữ liệu phục vụ cho các bài toán phân đoạn hình ảnh thực phẩm.
- **caltech-101:** Bộ dữ liệu tổng hợp các đối tượng khác nhau, được tận dụng như dữ liệu phi thực phẩm để cân bằng huấn luyện.
- **Food-5K:** Gồm 5.000 ảnh thực phẩm giúp mở rộng quy mô và tăng tính đa dạng cho tập huấn luyện.
- **CIFAR-100:** Dữ liệu phi thực phẩm lấy từ CIFAR-100 (link: `https://universe.roboflow.com/university-of-toronto-gzhju/cifar-10-3jsgm/dataset/2`), hỗ trợ quá trình huấn luyện với vai trò làm nhiễu dữ liệu đối lập.

Phụ lục B

Tổng hợp mã nguồn

1 Mã nguồn triển khai các mô hình

Các mô hình học sâu chính được xây dựng, huấn luyện và thử nghiệm thông qua các notebook và script, bao gồm:

- **Phân loại ảnh Food và Non-Food:** Mô hình CNN ResNet50 được tinh chỉnh để phân biệt ảnh chứa thực phẩm và ảnh không liên quan, truy cập tại đây [1]
- **Sinh caption cho ảnh món ăn:** Áp dụng kiến trúc BLIP để huấn luyện mô hình sinh chú thích ảnh chuyên biệt cho ảnh món ăn, truy cập tại đây. [2]
- **Sinh công thức nấu ăn từ ảnh:** Mô hình kết hợp xử lý ảnh và ngôn ngữ để dự đoán nguyên liệu và hướng dẫn chế biến từ ảnh đầu vào, truy cập tại đây. [3]

2 Mã nguồn hệ thống website

Toàn bộ mã nguồn phần giao diện và backend của hệ thống web bao gồm API, kết nối cơ sở dữ liệu, xử lý ảnh và hiển thị kết quả đã được đóng gói và công khai trên GitHub, truy cập tại đây. [4]

[1] https://www.kaggle.com/code/nguyenngoc7777/food-and-non-food-classification-resnet50/notebook
[2] https://www.kaggle.com/code/nguyenngoc7777/fine-tuning-blip-for-food
[3] https://www.kaggle.com/code/hungdangdinh/image2recipe-vietnamese
[4] https://github.com/anh7777/Visual-Captioning.git

Phụ lục C

Hướng dẫn sử dụng hệ thống

Phần phụ lục này nhằm cung cấp hướng dẫn chi tiết và dễ hiểu về cách sử dụng các tính năng chính của hệ thống tạo caption tự động cho hình ảnh. Người dùng có thể tham khảo để nắm bắt rõ quy trình thao tác, từ khâu đăng nhập đến việc quản lý nội dung đã tạo, nhằm khai thác hệ thống một cách hiệu quả và thuận tiện nhất.

1 Tổng quan hệ thống

Hệ thống được thiết kế với giao diện trực quan, thân thiện với người dùng. Các chức năng chính được tổ chức rõ ràng, hiển thị thông qua thanh công cụ bên trái (sidebar) và thanh thông tin phía trên (topbar). Cụ thể, người dùng có thể thực hiện các thao tác sau:

- Tạo tài khoản mới và đăng nhập để sử dụng đầy đủ các tính năng.
- Tải ảnh từ thiết bị cá nhân và tạo caption tự động.
- Lưu trữ, quản lý ảnh và caption trong các album cá nhân.
- Đánh giá chất lượng caption để phản hồi hoặc cải thiện hệ thống.
- Theo dõi số liệu sử dụng như dung lượng, tần suất truy cập...
- Gửi yêu cầu hỗ trợ nếu gặp sự cố hoặc cần tư vấn từ bộ phận quản trị.

2 Đăng ký và đăng nhập

2.1 Tạo tài khoản mới

Để bắt đầu sử dụng hệ thống, người dùng cần đăng ký tài khoản cá nhân theo các bước sau:

1. Truy cập vào trang chủ của hệ thống.
2. Tại giao diện đăng nhập, chọn liên kết đăng ký để chuyển sang trang đăng ký.
3. Nhập đầy đủ các thông tin cần thiết như tên đăng nhập, địa chỉ email, và mật khẩu.

4. Nhấn xác nhận để hoàn tất đăng ký.

Sau khi gửi biểu mẫu đăng ký, hệ thống sẽ gửi một email xác thực đến địa chỉ email đã cung cấp. Người dùng cần truy cập vào hộp thư và nhấp vào đường dẫn xác nhận để kích hoạt tài khoản.

2.2 Đăng nhập hệ thống

Khi đã có tài khoản, người dùng có thể đăng nhập nhanh chóng:

- Truy cập trang chủ.
- Nhập tên đăng nhập và mật khẩu đã đăng ký.
- Nhấn vào nút đăng nhập.

Nếu thông tin hợp lệ, hệ thống sẽ tự động chuyển đến giao diện làm việc chính của người dùng.

2.3 Quản lý tài khoản

Sau khi đăng nhập thành công, người dùng có thể điều chỉnh thông tin tài khoản cá nhân tại biểu tượng người dùng nằm ở góc phải phía trên màn hình. Các thao tác phổ biến bao gồm:

- Thay đổi mật khẩu.
- Cập nhật địa chỉ email.
- Đăng xuất khỏi hệ thống để đảm bảo bảo mật.

3 Tải hình ảnh và tạo caption

3.1 Truy cập chức năng tải ảnh

Người dùng có thể bắt đầu quá trình tạo caption bằng cách chọn biểu tượng mũi tên tải lên ở thanh bên trái. Tại đây, hệ thống sẽ cho phép người dùng tải ảnh lên và kích hoạt quá trình tạo caption.

3.2 Cách thức tải ảnh

Có hai phương thức để tải ảnh vào hệ thống:

- Kéo và thả: Chỉ cần kéo ảnh từ thư mục trên máy tính và thả vào khu vực tải lên trên giao diện.
- Chọn từ máy: Nhấn vào nút chọn tệp để mở hộp thoại chọn ảnh từ máy tính.

Hệ thống hỗ trợ các định dạng ảnh thông dụng như JPG, PNG, GIF và BMP. Mỗi ảnh không vượt quá 10MB để đảm bảo tốc độ xử lý.

3.3 Tạo caption tự động

Khi ảnh đã được tải thành công, hệ thống sẽ tự động xử lý để sinh caption:

- Một thanh tiến trình sẽ hiển thị trạng thái xử lý ảnh.
- Khi hoàn tất, hệ thống hiển thị ảnh đã tải và caption được tạo.
- Người dùng có thể xem trước nội dung caption ngay trên giao diện.

Đối với các ảnh là món ăn, hệ thống còn cung cấp thông tin nhận diện món ăn nếu có, như tên món, nguyên liệu dự đoán.

3.4 Lưu ảnh và caption vào album

Sau khi caption được sinh ra, người dùng có thể lưu nội dung vào bộ sưu tập cá nhân như sau:

- Nhấn vào biểu tượng lưu tại ảnh đã tạo caption.
- Chọn album sẵn có hoặc tạo một album mới.
- Xác nhận bằng cách nhấn nút lưu.

3.5 Đánh giá caption

Nếu người dùng thấy caption không phù hợp hoặc cần cải thiện, có thể gửi đánh giá cho hệ thống:

- Nhấn biểu tượng ngôi sao để mở giao diện đánh giá.
- Chọn số sao từ 1 đến 5 và nhập nhận xét.
- Nhấn nút gửi đánh giá để hoàn tất.

Mọi đánh giá đều được lưu trữ và phân tích để cải tiến chất lượng caption trong các phiên bản tiếp theo.

4 Quản lý album và hình ảnh

Toàn bộ ảnh và caption đã lưu sẽ được tổ chức theo album trong khu vực quản lý cá nhân. Tại đây, người dùng có thể:

- Tìm kiếm nhanh ảnh hoặc album theo từ khóa.
- Đổi tên, xóa ảnh hoặc album không cần thiết.
- Sắp xếp lại hình ảnh để dễ theo dõi hơn.

Tính năng này giúp quản lý dữ liệu dễ dàng, đặc biệt khi khối lượng ảnh lớn.

5 Xem thống kê sử dụng

Hệ thống cung cấp một bảng thống kê hiển thị dung lượng đã sử dụng, số lượng ảnh đã tải lên, số lần tạo caption, và các chỉ số hoạt động khác. Giao diện thống kê được trình bày bằng biểu đồ trực quan, giúp người dùng theo dõi và kiểm soát việc sử dụng hiệu quả hơn.

6 Gửi yêu cầu hỗ trợ

Trong trường hợp gặp vấn đề kỹ thuật hoặc cần giải đáp, người dùng có thể truy cập trang hỗ trợ để gửi yêu cầu. Chỉ cần điền thông tin vào biểu mẫu và mô tả sự cố đang gặp phải. Hệ thống sẽ ghi nhận và gửi phản hồi qua biểu tượng thông báo hoặc email sớm nhất có thể.

7 Tổng kết

Hệ thống sinh caption tự động cho hình ảnh được phát triển để hỗ trợ người dùng xử lý ảnh nhanh chóng, tiện lợi và linh hoạt. Các chức năng được thiết kế đồng bộ từ khâu tải ảnh, sinh caption, quản lý album đến đánh giá và hỗ trợ kỹ thuật. Người dùng nên sử dụng ảnh có chất lượng tốt, tổ chức album hợp lý và thường xuyên phản hồi để giúp hệ thống ngày càng hoàn thiện hơn.