

**224 ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN**

ĐỒ ÁN TỐT NGHIỆP

ĐỀ TÀI:

**XÂY DỰNG HỆ THỐNG QUẢN LÝ TỬ ĐỒ THÔNG
MINH SỬ DỤNG KỸ THUẬT DEEP LEARNING
VỚI KIẾN TRÚC MOBILENETV2**

Người hướng dẫn: **ThS. NGUYỄN THẾ XUÂN LY**

Sinh viên thực hiện: **NGUYỄN PHAN VIỆT ANH**

Số thẻ sinh viên: **102200202**

Lớp: **20TCLC_DT5**

Đà Nẵng, 2026

MỤC LỤC

TÓM TẮT.....	9
LỜI NÓI ĐẦU.....	11
LỜI CAM ĐOAN.....	13
DANH SÁCH CÁC TỪ VIẾT TẮT	15
DANH SÁCH HÌNH ẢNH.....	17
DANH SÁCH BẢNG:.....	18
CHƯƠNG 1: MỞ ĐẦU	19
1.1. Đặt vấn đề.....	19
1.2. Mục tiêu đề tài	20
1.3. Phạm vi nghiên cứu	21
1.4. Cấu trúc báo cáo.....	22
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ	23
2.1. Deep Learning và Mạng Nơ-ron Tích chập (CNN).....	23
2.1.1. Tổng quan về CNN (Convolutional Neural Networks)	23
2.1.2. Lớp Tích chập (Convolutional Layer)	23
2.1.3. Hàm kích hoạt phi tuyến (Activation Function - ReLU)	24
2.1.4. Lớp Gộp (Pooling Layer)	24
2.1.5. Lớp Kết nối đầy đủ (Fully Connected Layer - FC)	25
2.2. Kiến trúc MobileNetV2.....	25
2.2.1. Inverted Residuals (Khối thặng dư đảo ngược).....	25
2.2.2. Linear Bottlenecks	27
2.2.3. So sánh MobileNetV1 và MobileNetV2	29
2.3. Các công nghệ tích hợp	29
CHƯƠNG 3: THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG	30
3.1. Quy trình xử lý dữ liệu (Data Pipeline).....	30
3.1.1. Thu thập dữ liệu đa nguồn (Hybrid Data Collection).....	30
3.1.2. Tiền xử lý và Làm sạch dữ liệu (Data Cleaning)	31

3.1.3. Phân chia tập dữ liệu (Data Splitting)	31
3.1.4. Tăng cường dữ liệu (Data Augmentation).....	32
3.1.5. Chuẩn hóa đầu vào (Normalization).....	32
3.2. Kiến trúc hệ thống.....	32
3.3. Thiết kế Cơ sở dữ liệu (Database Schema)	34
3.3.1. Cấu trúc Collection "History"	35
3.3.2. Mẫu dữ liệu (Data Sample).....	36
3.3.3. Cơ chế lưu trữ vật lý & An toàn dữ liệu.....	36
3.4. Phân tích chức năng hệ thống (Use-case Description).....	37
3.5. Thiết kế luồng xử lý chi tiết (Sequence Diagram)	40
3.6. Các giải pháp bảo mật và an toàn hệ thống.....	41
3.6.1. Kiểm tra và xác thực dữ liệu đầu vào (Strict Input Validation) ..	41
3.6.2. Cơ chế chia sẻ tài nguyên nguồn gốc chéo (CORS Policies)	41
3.6.3. Làm sạch dữ liệu và Chống tấn công đường dẫn (Anti-Path Traversal)	42
3.6.4. Cô lập môi trường bằng Docker (Container Isolation).....	42
3.6.5. Quản lý cấu hình và Bí mật (Secrets Management)	43
CHƯƠNG 4: THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ (QUAN TRỌNG NHẤT).....	45
4.1. Xây dựng và Xử lý Bộ dữ liệu (Dataset Construction)	45
4.1.1. Chiến lược Dữ liệu lai (Hybrid Dataset).....	45
4.1.2. Quy trình Tiền xử lý (Preprocessing)	45
4.1.3. Thống kê dữ liệu	47
4.2. Thực nghiệm Lựa chọn Mô hình (Model Benchmarking)	48
4.2.1. Thiết lập thực nghiệm	48
4.2.2. Kết quả so sánh	49

4.2.3. Phân tích và Lựa chọn mô hình tối ưu cho đề án	50
4.2.4. Phân tích thực nghiệm và lựa chọn tham số tối ưu (Hyperparameter Analysis)	50
a. Thiết lập kịch bản thực nghiệm.....	51
b. Kết quả thực nghiệm và đánh giá	52
c. Biện luận và quyết định lựa chọn tham số.....	53
4.3. Đánh giá Kết quả Huấn luyện Chuyên sâu (Final Evaluation)	54
4.3.1. Hiệu năng tổng thể.....	55
4.3.2. Phân tích Quá trình Huấn luyện (Training History Analysis).....	55
4.3.3. Phân tích chi tiết từng lớp (Classification Report Analysis)	56
4.3.4. Phân tích Lỗi qua Ma trận Nhầm lẫn (Confusion Matrix Analysis)	57
4.4. Tối ưu hóa triển khai: Xử lý "Ảo giác AI" (AI Hallucination).....	60
4.5. So sánh với các nghiên cứu và ứng dụng tương tự.....	62
4.6. Demo Sản phẩm.....	63
4.7. Đánh giá hiệu năng và Trải nghiệm người dùng (System Performance Evaluation)	64
4.7.1. Thiết lập môi trường và Phương pháp đo đạc	64
4.7.2. Phân tích độ trễ chi tiết (Latency Breakdown).....	65
4.7.3. Đánh giá Trải nghiệm người dùng (UX).....	66
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	67
5.1. Tổng kết các kết quả đạt được	67
5.2. Đánh giá và Các hạn chế tồn tại.....	68
5.3. Hướng phát triển trong tương lai	69
TÀI LIỆU THAM KHẢO	71

NHẬN XÉT ĐỒ ÁN TỐT NGHIỆP

1. Thông tin chung:

1. Họ và tên sinh viên: Nguyễn Phan Việt Anh
2. Lớp: 20TCLC_DT5 Số thẻ SV: 102200202
3. Tên đề tài: **XÂY DỰNG HỆ THỐNG QUẢN LÝ TỬ ĐỒ THÔNG MINH SỬ DỤNG KỸ THUẬT DEEP LEARNING VỚI KIẾN TRÚC MOBILENETV2**
4. Người hướng dẫn: **NGUYỄN THẾ XUÂN LY** Học hàm/ học vị: **Thạc Sĩ**

II. Nhận xét, đánh giá đồ án tốt nghiệp:

1. Về tính cấp thiết, tính mới, khả năng ứng dụng của đề tài: (điểm tối đa là 2đ)
.....
.....
2. Về kết quả giải quyết các nội dung nhiệm vụ yêu cầu của đồ án:
(điểm tối đa là 4đ)
.....
.....
3. Về hình thức, cấu trúc, bố cục của đồ án tốt nghiệp: (điểm tối đa là 2đ)
.....
.....
4. Đề tài có giá trị khoa học/ có bài báo/ giải quyết vấn đề đặt ra của doanh nghiệp hoặc nhà trường: (điểm tối đa là 1đ)
.....
.....
5. Các tồn tại, thiếu sót cần bổ sung, chỉnh sửa:
.....
.....
.....

III. Tinh thần, thái độ làm việc của sinh viên: (điểm tối đa 1đ)

.....
.....

IV. Đánh giá:

1. Điểm đánh giá:/10 (lấy đến 1 số lẻ thập phân)

2. Đề nghị: Được bảo vệ đồ án Bổ sung đề bảo vệ

Không được bảo vệ

Đà Nẵng, ngày tháng năm 202

Người hướng dẫn

NHẬN XÉT PHẢN BIỆN ĐỒ ÁN TỐT NGHIỆP

I. Thông tin chung:

1. Họ và tên sinh viên: Nguyễn Phan Việt Anh
2. Lớp: 20TCLC_DT5 Số thẻ SV: 102200202
3. Tên đề tài: **XÂY DỰNG HỆ THỐNG QUẢN LÝ TỬ ĐỒ THÔNG MINH SỬ DỤNG KỸ THUẬT DEEP LEARNING VỚI KIẾN TRÚC MOBILENETV2**
4. Người phản biện: **Đông Ngọc Nguyên Thịnh** Học hàm/ học vị: **Thạc Sĩ**

II. Nhận xét, đánh giá đồ án tốt nghiệp:

TT	Các tiêu chí đánh giá	Điểm tối đa	Điểm đánh giá
1	Sinh viên có phương pháp nghiên cứu phù hợp, giải quyết đủ nhiệm vụ đồ án được giao	80	
1a	- Tính mới (nội dung chính của ĐATN có những phần mới so với các ĐATN trước đây). - Đề tài có giá trị khoa học, công nghệ; có thể ứng dụng thực tiễn.	15	
1b	- Kỹ năng giải quyết vấn đề; hiểu, vận dụng được kiến thức cơ bản, cơ sở, chuyên ngành trong vấn đề nghiên cứu. - Chất lượng nội dung ĐATN (thuyết minh, bản vẽ, chương trình, mô hình,).	50	
1c	- Có kỹ năng vận dụng thành thạo các phần mềm ứng dụng trong vấn đề nghiên cứu. - Có kỹ năng đọc, hiểu tài liệu bằng tiếng nước ngoài ứng dụng trong vấn đề nghiên cứu. - Có kỹ năng làm việc nhóm;	15	

2	Kỹ năng viết:	20	
2a	- Bố cục hợp lý, lập luận rõ ràng, chặt chẽ, lời văn súc tích	15	
2b	- Thuyết minh đề án không có lỗi chính tả, in ấn, định dạng	5	
3	Tổng điểm đánh giá theo thang 100:		
	Quy về thang 10 (lấy đến 1 số lẻ)		

1. Các tồn tại, thiếu sót cần bổ sung, chỉnh sửa:

.....
.....
.....

2. Câu hỏi đề nghị sinh viên trả lời trong buổi bảo vệ:

.....
.....
.....

3. Đề nghị: Được bảo vệ đề án Bổ sung đề bảo vệ Không được
bảo vệ

Đà Nẵng, ngày 23 tháng 01 năm 2026

Người phản biện

TÓM TẮT

Tên đề tài: Xây dựng hệ thống quản lý tủ đồ thông minh sử dụng kỹ thuật Deep Learning với kiến trúc MobileNetV2

Sinh viên thực hiện: Nguyễn Phan Việt Anh

Số thẻ SV: 102200202 **Lớp:** 20TCLC_DT5

Trong bối cảnh công nghệ 4.0 và sự bùng nổ của thị giác máy tính (Computer Vision), việc ứng dụng Trí tuệ nhân tạo vào đời sống thường nhật đang trở thành xu hướng tất yếu. Đề án này tập trung nghiên cứu và phát triển một hệ thống **Quản lý tủ đồ thông minh (Smart Wardrobe)**, đóng vai trò như một trợ lý ảo giúp người dùng số hóa và quản lý trang phục cá nhân, giải quyết bài toán "lãng phí thời gian lựa chọn trang phục" thông qua tự động hóa.

Về mặt thuật toán, đề án đề xuất sử dụng kỹ thuật **Học sâu (Deep Learning)** với kiến trúc mạng nơ-ron tích chập **MobileNetV2**. Đây là mô hình được tối ưu hóa đặc biệt cho các thiết bị biên và di động nhờ cấu trúc *Inverted Residuals* và *Linear Bottlenecks*, giúp giảm thiểu chi phí tính toán nhưng vẫn duy trì hiệu năng cao. Chiến lược **Chuyển giao tri thức (Transfer Learning)** được áp dụng để tinh chỉnh mô hình, giúp hệ thống nhanh chóng thích nghi với các đặc trưng của trang phục từ bộ trọng số ImageNet.

Về mặt dữ liệu, đề án đã xây dựng thành công bộ dữ liệu quy mô lớn lên tới hơn **131.000 hình ảnh** thuộc 20 danh mục trang phục phổ biến (như Áo thun, Polo, Quần Jean, Váy, Giày...). Quy trình xử lý dữ liệu áp dụng chiến lược **Dữ liệu lai (Hybrid Dataset)**: kết hợp giữa nguồn dữ liệu chuẩn (Kaggle) và dữ liệu thực tế thu thập từ Internet, cùng với các kỹ thuật tăng cường dữ liệu (Data Augmentation) để nâng cao khả năng tổng quát hóa của mô hình.

Về mặt kỹ thuật hệ thống, ứng dụng được xây dựng theo kiến trúc **Microservices** hiện đại và đóng gói bằng công nghệ **Docker Container**, đảm bảo khả năng triển

khai linh hoạt và đồng nhất trên mọi môi trường. Hệ thống bao gồm Backend xử lý AI (Flask API), Frontend giao diện người dùng (ReactJS) và Cơ sở dữ liệu phi quan hệ (MongoDB) để lưu trữ metadata hình ảnh. Bên cạnh chức năng nhận diện cốt lõi, hệ thống tích hợp thuật toán **Gợi ý theo ngữ cảnh (Context-aware Recommendation)** dựa trên dữ liệu thời tiết thời gian thực (lấy từ OpenWeatherMap API) và cơ chế **Kiểm soát ngưỡng tin cậy (Confidence Threshold)** để lọc bỏ nhiễu, đảm bảo độ chính xác cao nhất cho người dùng.

Kết quả thực nghiệm trên tập kiểm thử độc lập cho thấy mô hình MobileNetV2 sau khi tinh chỉnh đạt độ chính xác **~86%**, với tốc độ suy luận trung bình chỉ **22ms/ảnh**. Hệ thống hoạt động ổn định, giao diện trực quan, đáp ứng tốt yêu cầu về thời gian thực (Real-time) và mang lại trải nghiệm quản lý thời trang hiệu quả, khoa học cho người dùng.

LỜI NÓI ĐẦU

Trong bối cảnh toàn cầu hóa và sự bùng nổ mạnh mẽ của cuộc Cách mạng công nghiệp lần thứ tư (Industry 4.0), công nghệ thông tin không còn đơn thuần là công cụ hỗ trợ mà đã trở thành nền tảng cốt lõi định hình lại mọi mặt của đời sống xã hội. Đặc biệt, sự trỗi dậy của Trí tuệ nhân tạo (Artificial Intelligence - AI) và Học sâu (Deep Learning) đang tạo ra những bước đột phá chưa từng có, chuyển dịch từ các phòng thí nghiệm hàn lâm sang các ứng dụng thực tiễn, phục vụ trực tiếp cho nhu cầu của con người từ y tế, giáo dục, giao thông cho đến các tiện ích sinh hoạt thường nhật.

Thời trang, một trong những nhu cầu thiết yếu nhất của con người, cũng không nằm ngoài dòng chảy công nghệ này. Tuy nhiên, sự phát triển của ngành công nghiệp "Thời trang nhanh" (Fast Fashion) đang đặt ra những thách thức mới. Người tiêu dùng hiện đại sở hữu số lượng trang phục ngày càng lớn, dẫn đến nghịch lý "Tủ đồ đầy ắp nhưng không có gì để mặc". Việc quản lý tủ quần áo trở thành một gánh nặng vô hình, tiêu tốn thời gian và năng lượng tinh thần mỗi ngày. Hơn nữa, việc mua sắm không kiểm soát và lãng phí trang phục cũng đang là một vấn đề nhức nhối đối với môi trường.

Xuất phát từ những trăn trở đó, tôi đã lựa chọn và thực hiện đề án tốt nghiệp với đề tài "**Xây dựng hệ thống quản lý tủ đồ thông minh sử dụng kỹ thuật Deep Learning với kiến trúc MobileNetV2**". Đề tài không chỉ dừng lại ở mục tiêu giải quyết bài toán phân loại ảnh đơn thuần, mà hướng tới việc xây dựng một "Trợ lý ảo" thực thụ, giúp số hóa tủ quần áo, tự động hóa quy trình quản lý và cung cấp các gợi ý phối đồ thông minh dựa trên ngữ cảnh thời tiết thực tế.

Quá trình thực hiện đề án là một hành trình nghiên cứu đầy thách thức nhưng cũng vô cùng thú vị. Đây là cơ hội quý báu để tôi tổng hợp và vận dụng toàn bộ kiến thức đã tích lũy trong suốt 5 năm học tập tại trường, từ các nền tảng cơ bản về Công nghệ phần mềm, Cơ sở dữ liệu đến các kỹ thuật chuyên sâu như Thị giác máy tính (Computer Vision) và Học máy (Machine Learning).

Thông qua đồ án này, tôi đã có cơ hội đi sâu nghiên cứu về kiến trúc mạng nơ-ron tích chập MobileNetV2 – một giải pháp tối ưu cho các thiết bị biên, giúp cân bằng giữa độ chính xác và tốc độ xử lý. Bên cạnh đó, đồ án cũng là môi trường thực nghiệm để tôi rèn luyện kỹ năng xây dựng hệ thống phần mềm hoàn chỉnh (Full-stack Development), từ việc thiết kế kiến trúc Microservices với Docker, xây dựng API với Flask đến việc triển khai cơ sở dữ liệu phi quan hệ MongoDB. Những khó khăn trong quá trình thu thập dữ liệu, tinh chỉnh tham số mô hình hay xử lý các vấn đề về hiệu năng hệ thống đã mang lại cho tôi những bài học kinh nghiệm vô giá cho con đường nghề nghiệp sau này.

Mặc dù đã nỗ lực hết mình để hoàn thiện hệ thống, nhưng do giới hạn về thời gian và kinh nghiệm thực tế, đồ án chắc chắn không tránh khỏi những thiếu sót. Tôi rất mong nhận được những ý kiến đóng góp quý báu từ Quý Thầy Cô và các bạn để đề tài có thể được phát triển hoàn thiện hơn nữa trong tương lai.

Tôi hy vọng rằng, kết quả của đồ án này sẽ là một bước đệm nhỏ, góp phần minh chứng cho tiềm năng ứng dụng của AI trong việc nâng cao chất lượng cuộc sống cá nhân (Personalized AI), biến những công nghệ phức tạp trở nên gần gũi và hữu ích cho mọi người.

LỜI CAM ĐOAN

Tôi tên là: **Nguyễn Phan Việt Anh** Mã số sinh viên: **102200202** Sinh viên lớp: **20TCLC_DT5** Trường Đại học Bách Khoa - Đại học Đà Nẵng

Tôi xin cam đoan rằng đồ án tốt nghiệp với đề tài "**Xây dựng hệ thống quản lý tủ đồ thông minh sử dụng kỹ thuật Deep Learning với kiến trúc MobileNetV2**" là công trình nghiên cứu, tìm hiểu và thực hiện độc lập của riêng bản thân tôi, dưới sự hướng dẫn khoa học tận tình của Giảng viên **ThS. Nguyễn Thế Xuân Ly**.

Tôi xin khẳng định và chịu trách nhiệm về tính trung thực của các nội dung sau:

1. **Về dữ liệu:** Bộ dữ liệu hình ảnh thời trang (Hybrid Dataset) sử dụng trong đồ án được tôi tự xây dựng thông qua quy trình thu thập từ các nguồn dữ liệu mở (Kaggle) và sử dụng thuật toán để cào dữ liệu (Crawling) từ các trang web thực tế. Quy trình làm sạch, gán nhãn và tăng cường dữ liệu được thực hiện minh bạch, không sao chép nguyên bản từ bất kỳ bộ dữ liệu có sẵn nào khác.
2. **Về giải pháp công nghệ:** Mã nguồn chương trình (Source Code), bao gồm việc xây dựng mô hình MobileNetV2, các script xử lý dữ liệu, API Backend (Flask), Giao diện Frontend và cấu hình hệ thống Docker là sản phẩm trí tuệ do chính tôi viết và phát triển. Các thư viện mã nguồn mở được sử dụng đều được tuân thủ theo giấy phép sử dụng tương ứng.
3. **Về kết quả thực nghiệm:** Các số liệu về độ chính xác (Accuracy), độ trễ (Latency) và các bảng so sánh hiệu năng mô hình được trình bày trong đồ án là kết quả thực tế từ quá trình huấn luyện và kiểm thử trên máy tính cá nhân, tuyệt đối không có sự ngụy tạo hay sửa đổi số liệu để làm đẹp kết quả.

4. **Về tài liệu tham khảo:** Tất cả các nguồn tài liệu, lý thuyết, công thức và ý tưởng tham khảo từ các sách, báo, tạp chí khoa học, và các trang web uy tín đều được trích dẫn nguồn gốc rõ ràng, đầy đủ trong danh mục Tài liệu tham khảo theo đúng quy định.

Tôi cam kết tuyệt đối không sao chép nguyên văn từ bất kỳ đồ án tốt nghiệp nào trước đó. Tôi xin chịu hoàn toàn trách nhiệm về tính liêm chính học thuật của đồ án này. Nếu phát hiện có bất kỳ sự gian lận hay vi phạm quy định về quyền sở hữu trí tuệ, tôi xin chấp nhận mọi hình thức kỷ luật của Hội đồng bảo vệ đồ án và Nhà trường, kể cả việc hủy bỏ kết quả tốt nghiệp.

Sinh viên thực hiện

Nguyễn Phan Việt Anh

DANH SÁCH CÁC TỪ VIẾT TẮT

Từ viết tắt	Nguyên nghĩa (Tiếng Anh)	Nghĩa tiếng Việt
AI	Artificial Intelligence	Trí tuệ nhân tạo
API	Application Programming Interface	Giao diện lập trình ứng dụng
CNN	Convolutional Neural Network	Mạng nơ-ron tích chập
CORS	Cross-Origin Resource Sharing	Chia sẻ tài nguyên chéo nguồn (Cơ chế bảo mật Web)
CPU	Central Processing Unit	Đơn vị xử lý trung tâm
CSDL	Database	Cơ sở dữ liệu
DL	Deep Learning	Học sâu
DoS	Denial of Service	Tấn công từ chối dịch vụ
GPU	Graphics Processing Unit	Đơn vị xử lý đồ họa
HTML	Hypertext Markup Language	Ngôn ngữ đánh dấu siêu văn bản
HTTP	Hypertext Transfer Protocol	Giao thức truyền tải siêu văn bản
IDE	Integrated Development Environment	Môi trường phát triển tích hợp
JSON	JavaScript Object Notation	Định dạng dữ liệu đối tượng JavaScript
MD5	Message-Digest Algorithm 5	Thuật toán mã hóa băm (dùng để lọc trùng lặp)
NoSQL	Not Only SQL	Cơ sở dữ liệu phi quan hệ (Ví dụ: MongoDB)
RAM	Random Access Memory	Bộ nhớ truy cập ngẫu nhiên
ReLU	Rectified Linear Unit	Hàm kích hoạt tuyến tính chỉnh lưu
RGB	Red Green Blue	Hệ màu cơ bản (Đỏ - Lục - Lam)

SGD	Stochastic Gradient Descent	Thuật toán tối ưu hóa Gradient ngẫu nhiên
SSL	Secure Sockets Layer	Lớp công bảo mật (Mã hóa truyền tin)
TMDT	E-commerce	Thương mại điện tử
UI	User Interface	Giao diện người dùng
URL	Uniform Resource Locator	Định vị tài nguyên thống nhất (Đường dẫn Web)
UUID	Universally Unique Identifier	Mã định danh duy nhất toàn cầu
UX	User Experience	Trải nghiệm người dùng
VRAM	Video Random Access Memory	Bộ nhớ đồ họa (Bộ nhớ của GPU)

DANH SÁCH HÌNH ẢNH

Hình 2.2.1.1 Cấu trúc khối Inverted Residual trong MobileNetV2 (Hẹp => Rộng => Hẹp).	26
Hình 3.2.1 Sơ đồ kiến trúc tổng thể của hệ thống.....	33
Hình 3.4.1 Biểu đồ ca sử dụng (Use-case) của hệ thống.	37
Hình 3.5.1 Biểu đồ tuần tự xử lý luồng nhận diện và gợi ý.....	40
Hình 4.1.2.1 Ảnh trước và sau khi biến đổi.	46
Hình 4.1.3.1 Bảng Thống Kê Dữ Liệu.....	47
Hình 4.2.2.1 Biểu đồ so sánh Accuracy qua các Epochs đầu tiên của 3 mô hình	49
Hình 4.2.4.1 Biểu đồ so sánh quá trình hội tụ của mô hình MobileNetV2 giữa các kịch bản siêu tham số.	52
Hình 4.3.2.1 Biểu đồ quá trình huấn luyện:.....	55
Hình 4.3.4.1 Ma trận nhầm lẫn trên tập dữ liệu kiểm thử.....	57
Hình 4.4.1 Nhầm lẫn gổ và túi xách	60
Hình 4.4.2 Không xác định vật thể không phải áo quần và phụ kiện thời trang..	61
Hình 4.6.1 Giao diện nhận diện thành công váy và gợi ý thời tiết	63

DANH SÁCH BẢNG:

Bảng 2.2.3.1 So sánh MoblieNetV1 và MoblieNetV2	29
Bảng 3.3.1.1 Cấu trúc dữ liệu	35
Bảng 4.1.2.1 Các kỹ thuật tăng cường dữ liệu (Data Augmentation) áp dụng....	46
Bảng 4.2.2.1 So sánh độ hiệu quả của các mô hình.....	49
Bảng 4.2.4.2 So sánh kết quả thực nghiệm giữa các kịch bản siêu tham số.....	52
Bảng 4.3.1 Bảng Báo cáo Phân loại chi tiết.....	54
Bảng 4.3.4.1 Top 5 trường hợp nhầm lẫn phổ biến và Giải pháp đề xuất	58
Bảng 4.5.1 So sánh đề tài với các nghiên cứu/ứng dụng hiện có	62
Bảng 4.7.2.1 Bảng phân tích thời gian xử lý trung bình.....	65

CHƯƠNG 1: MỞ ĐẦU

1.1. Đặt vấn đề

Trong kỷ nguyên Cách mạng công nghiệp 4.0, Trí tuệ nhân tạo (Artificial Intelligence - AI) và Học sâu (Deep Learning) không còn là những khái niệm hàn lâm nằm trong phòng thí nghiệm, mà đã len lỏi vào từng ngóc ngách của đời sống thường nhật. Đặc biệt, lĩnh vực Thị giác máy tính (Computer Vision) đã có những bước tiến vượt bậc, cho phép máy móc "nhìn" và "hiểu" hình ảnh với độ chính xác tiệm cận, thậm chí vượt qua khả năng của con người trong một số tác vụ cụ thể. Sự bùng nổ của các thiết bị di động thông minh cùng năng lực tính toán ngày càng mạnh mẽ đã mở ra cơ hội to lớn để đưa các mô hình AI phức tạp đến gần hơn với người dùng cuối.

Bên cạnh sự phát triển của công nghệ, ngành công nghiệp thời trang cũng đang chứng kiến sự thay đổi chóng mặt với xu hướng "Thời trang nhanh" (Fast Fashion). Người tiêu dùng hiện đại, đặc biệt là giới trẻ, sở hữu số lượng trang phục ngày càng lớn. Tuy nhiên, điều này dẫn đến một nghịch lý phổ biến: "Tủ đồ đầy ắp nhưng không có gì để mặc". Nhiều nghiên cứu chỉ ra rằng, con người thường chỉ sử dụng khoảng 20-30% số lượng quần áo mình sở hữu, trong khi phần còn lại bị lãng quên ở đáy tủ. Điều này không chỉ gây lãng phí về mặt tài chính cá nhân mà còn tạo ra gánh nặng lớn cho môi trường khi rác thải thời trang đang trở thành vấn đề nhức nhối toàn cầu. Ngoài ra, việc mất thời gian mỗi sáng để suy nghĩ "hôm nay mặc gì", phối đồ ra sao cho phù hợp với thời tiết và hoàn cảnh cũng là một vấn đề gây lãng phí thời gian và năng lượng tinh thần đáng kể.

Xuất phát từ thực tế đó, nhu cầu về một công cụ hỗ trợ quản lý trang phục cá nhân một cách khoa học, thông minh và tự động hóa là vô cùng cấp thiết. Một hệ thống lý tưởng không chỉ đơn thuần là nơi lưu trữ hình ảnh quần áo, mà phải đóng vai trò như một "Stylist ảo": có khả năng tự động nhận diện loại trang phục, phân loại chúng, và đưa ra các gợi ý phối đồ dựa trên các yếu tố ngoại cảnh như thời tiết.

Tuy nhiên, việc triển khai các mô hình Deep Learning trên các ứng dụng người dùng cuối gặp phải rào cản lớn về tài nguyên phần cứng. Các mô hình CNN truyền thống như VGG16 hay ResNet50 dù có độ chính xác cao nhưng lại quá nặng nề và tốc độ xử lý chậm để chạy trên các thiết bị cá nhân (Local deployment). Bài toán đặt ra là làm thế nào để cân bằng giữa độ chính xác (Accuracy) và tốc độ xử lý (Latency) để tạo ra trải nghiệm người dùng mượt mà.

Để giải quyết bài toán trên, đề án "**Xây dựng hệ thống quản lý tủ đồ thông minh sử dụng kỹ thuật Deep Learning với kiến trúc MobileNetV2**" được thực hiện. Đề tài tập trung nghiên cứu và ứng dụng kiến trúc mạng MobileNetV2 – một mô hình được tối ưu hóa đặc biệt cho các thiết bị di động với cấu trúc "Inverted Residuals", giúp giảm thiểu đáng kể số lượng tham số mà vẫn duy trì độ chính xác cao.

1.2. Mục tiêu đề tài

Mục tiêu tổng quát của đề án là xây dựng một hệ thống phần mềm hoàn chỉnh (End-to-End System) hỗ trợ người dùng quản lý tủ quần áo cá nhân, tự động hóa việc phân loại và gợi ý trang phục thông qua ứng dụng công nghệ Deep Learning.

Các mục tiêu cụ thể bao gồm:

- **Nghiên cứu lý thuyết:** Tìm hiểu sâu về Mạng nơ-ron tích chập (CNN), kỹ thuật Transfer Learning và đặc biệt là kiến trúc MobileNetV2.
- **Xây dựng bộ dữ liệu:** Thu thập và xử lý bộ dữ liệu hình ảnh thời trang đa dạng (bao gồm dữ liệu từ Kaggle và dữ liệu thực tế tại Việt Nam), áp dụng các kỹ thuật tăng cường dữ liệu (Data Augmentation) để nâng cao khả năng tổng quát hóa của mô hình.
- **Huấn luyện và Tối ưu hóa mô hình:** Thực hiện huấn luyện (Fine-tuning) mô hình MobileNetV2, tinh chỉnh các siêu tham số (Hyperparameters) như Learning Rate, Dropout để đạt được sự cân bằng tối ưu giữa độ chính xác và tốc độ.

- **Xây dựng ứng dụng:** Phát triển hệ thống theo kiến trúc Microservices sử dụng Docker, bao gồm Backend (Flask API), Frontend (React/Web Interface) và Database (MongoDB) để tạo ra sản phẩm có khả năng triển khai thực tế.

1.3. Phạm vi nghiên cứu

a. Phạm vi về công nghệ:

- Sử dụng ngôn ngữ Python và thư viện TensorFlow/Keras để xây dựng mô hình AI.
- Sử dụng kiến trúc MobileNetV2 làm nền tảng (Backbone) cho bài toán phân loại ảnh.
- Sử dụng công nghệ Container hóa (Docker) để đóng gói và triển khai hệ thống.
- Sử dụng MongoDB (NoSQL) để lưu trữ metadata và lịch sử hoạt động.

b. Phạm vi về dữ liệu và nghiệp vụ:

- Hệ thống tập trung nhận diện và phân loại **20 loại trang phục** phổ biến nhất (Ví dụ: Áo thun, Áo sơ mi, Quần Jean, Váy, Giày...).
- Hệ thống cung cấp chức năng gợi ý phối đồ dựa trên nhiệt độ/thời tiết hiện tại (thông qua tích hợp API thời tiết).

c. Phạm vi về đối tượng sử dụng và triển khai (Giả định hệ thống):

- Đồ án được thiết kế định hướng là một ứng dụng **Trợ lý cá nhân (Personal Assistant)** hoạt động trên môi trường cục bộ (Self-hosted/Local Environment).
- **Mô hình triển khai:** Hệ thống chạy trực tiếp trên thiết bị của người dùng thông qua nền tảng Docker, thay vì mô hình SaaS (Software as a Service) tập trung.

- **Về bảo mật và xác thực:** Do đặc thù chạy cục bộ, dữ liệu hình ảnh được lưu trữ trực tiếp trên ổ cứng vật lý của người dùng (thông qua cơ chế Docker Volume mapping), quyền riêng tư được đảm bảo bởi chính thiết bị vật lý. Do đó, trong phạm vi đề án này, các chức năng xác thực phức tạp (Authentication/Login) được lược bỏ để tập trung tối đa tài nguyên nghiên cứu vào việc tối ưu hóa thuật toán lõi AI.

1.4. Cấu trúc báo cáo

Báo cáo đề án được trình bày trong 5 chương với nội dung chính như sau:

- **Chương 1: Mở đầu:** Trình bày tổng quan về bối cảnh, lý do chọn đề tài, mục tiêu và phạm vi nghiên cứu.
- **Chương 2: Cơ sở lý thuyết và Công nghệ:** Giới thiệu nền tảng về Deep Learning, mạng CNN, phân tích kiến trúc MobileNetV2 và các công nghệ sử dụng (Docker, MongoDB).
- **Chương 3: Phân tích và Thiết kế hệ thống:** Mô tả kiến trúc tổng thể Microservices, thiết kế cơ sở dữ liệu và quy trình nghiệp vụ.
- **Chương 4: Thực nghiệm và Đánh giá:** Trình bày quy trình chuẩn bị dữ liệu, quá trình huấn luyện, các thực nghiệm so sánh và kết quả tối ưu hóa tham số.
- **Chương 5: Kết luận và Hướng phát triển:** Tóm tắt kết quả đạt được, demo sản phẩm và đề xuất các hướng cải tiến trong tương lai.

Thông qua đề án này, em hy vọng đóng góp một giải pháp công nghệ nhỏ bé nhưng thiết thực, giúp người dùng quản lý cuộc sống cá nhân hiệu quả hơn trong kỷ nguyên số.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ

2.1. Deep Learning và Mạng Nơ-ron Tích chập (CNN)

2.1.1. Tổng quan về CNN (Convolutional Neural Networks)

Mạng Nơ-ron Tích chập (Convolutional Neural Networks - CNN) là một trong những kiến trúc Deep Learning chuyên biệt và hiệu quả nhất cho bài toán xử lý dữ liệu dạng lưới, đặc biệt là hình ảnh. Khác với mạng nơ-ron truyền thống (MLP) yêu cầu làm phẳng (flatten) dữ liệu đầu vào làm mất đi thông tin không gian, CNN duy trì cấu trúc không gian của ảnh, cho phép mô hình học được các đặc trưng từ mức thấp (cạnh, góc) đến mức cao (mắt, mũi, vật thể) thông qua các lớp tích chập.

Kiến trúc cơ bản của một mạng CNN bao gồm 3 thành phần chính: Lớp tích chập (Convolutional Layer), Lớp gộp (Pooling Layer) và Lớp kết nối đầy đủ (Fully Connected Layer).

2.1.2. Lớp Tích chập (Convolutional Layer)

Đây là thành phần cốt lõi của mạng CNN, chịu trách nhiệm trích xuất đặc trưng (Feature Extraction).

- **Cơ chế hoạt động:** Lớp này sử dụng các bộ lọc (Filters/Kernels) là các ma trận nhỏ (thường kích thước 3x3 hoặc 5x5) trượt qua toàn bộ bức ảnh đầu vào. Tại mỗi vị trí, bộ lọc thực hiện phép nhân chập (element-wise multiplication) với các pixel của ảnh và tính tổng để tạo ra một giá trị mới trên bản đồ đặc trưng (Feature Map).
- **Vai trò:** Các bộ lọc khác nhau sẽ học cách nhận diện các đặc trưng khác nhau. Ví dụ: các lớp đầu tiên học phát hiện cạnh ngang, cạnh dọc; các lớp sâu hơn học phát hiện các hình khối phức tạp hơn.

- **Các tham số chính:**

- *Kernel Size*: Kích thước bộ lọc (ví dụ 3x3).
- *Stride (Bước nhảy)*: Khoảng cách trượt của bộ lọc. Stride càng lớn thì Feature Map đầu ra càng nhỏ.
- *Padding (Lê)*: Thêm các pixel giá trị 0 bao quanh ảnh để giữ nguyên kích thước ảnh sau khi tích chập.

2.1.3. Hàm kích hoạt phi tuyến (Activation Function - ReLU)

Ngay sau mỗi lớp tích chập thường là một hàm kích hoạt phi tuyến. Trong đồ án này, hàm **ReLU (Rectified Linear Unit)** được sử dụng phổ biến.

- **Công thức:** $f(x) = \max(0, x)$.
- **Vai trò:** Giúp mô hình học được các mối quan hệ phức tạp (phi tuyến tính) trong dữ liệu, đồng thời giảm thiểu vấn đề triệt tiêu đạo hàm (vanishing gradient), giúp quá trình huấn luyện hội tụ nhanh hơn.

2.1.4. Lớp Gộp (Pooling Layer)

Lớp Pooling thường được đặt xen kẽ giữa các lớp tích chập để giảm kích thước không gian (chiều rộng và chiều cao) của Feature Map.

- **Max Pooling:** Lấy giá trị lớn nhất trong vùng cửa sổ trượt. Đây là phương pháp phổ biến nhất vì nó giữ lại các đặc trưng nổi bật nhất của ảnh.
- **Vai trò:**
 - Giảm số lượng tham số tính toán, giúp mô hình nhẹ hơn.
 - Kiểm soát hiện tượng quá khớp (Overfitting).
 - Tạo ra tính bất biến dịch chuyển (Translation Invariance) - giúp mô hình nhận diện được vật thể dù nó nằm lệch đi một chút trong khung hình.

2.1.5. Lớp Kết nối đầy đủ (Fully Connected Layer - FC)

Sau khi đi qua nhiều lớp Tích chập và Pooling, các bản đồ đặc trưng 2D/3D được làm phẳng (Flatten) thành một vector 1D và đưa vào mạng nơ-ron truyền thống (Fully Connected Layer).

- **Vai trò:** Lớp này đóng vai trò như một bộ phân loại (Classifier). Nó tổng hợp tất cả các đặc trưng cục bộ đã trích xuất được để đưa ra quyết định cuối cùng.
- **Lớp đầu ra (Output Layer):** Đối với bài toán phân loại đa lớp (20 loại trang phục), lớp cuối cùng sử dụng hàm kích hoạt **Softmax** để chuyển đổi các giá trị đầu ra thành xác suất dự đoán cho từng lớp (tổng các xác suất bằng 1).

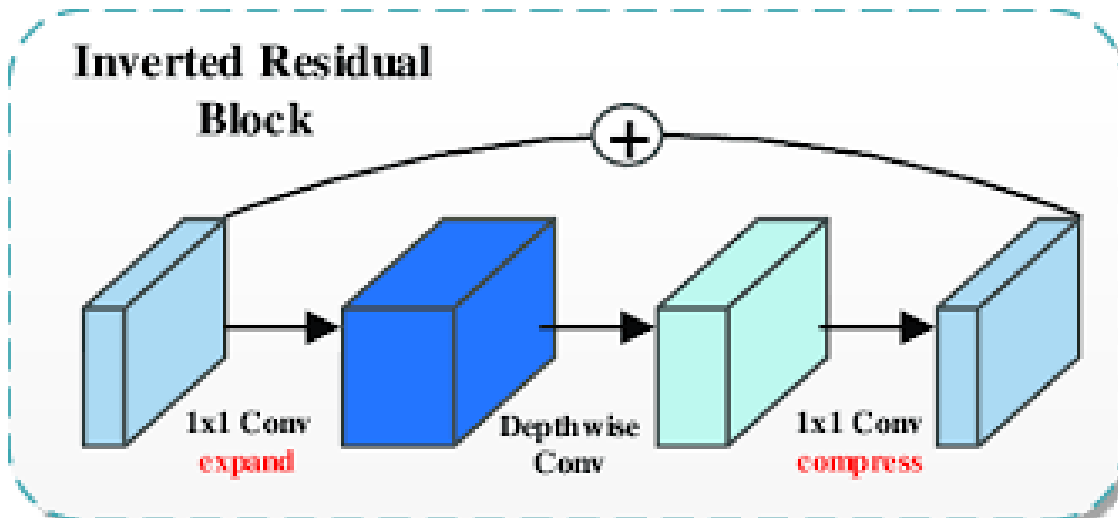
2.2. Kiến trúc MobileNetV2

MobileNetV2 (Sandler et al., 2018) là phiên bản cải tiến của MobileNetV1, được thiết kế nhằm tối ưu cho các bài toán thị giác máy tính trên thiết bị di động và hệ thống nhúng có tài nguyên tính toán hạn chế. Kiến trúc MobileNetV2 tập trung vào việc giảm số lượng tham số và chi phí tính toán trong khi vẫn duy trì độ chính xác cao. Mô hình này được xây dựng dựa trên hai thành phần cốt lõi: **Inverted Residuals** và **Linear Bottlenecks**.

2.2.1. Inverted Residuals (Khối thặng dư đảo ngược)

Khác với các kiến trúc ResNet truyền thống, nơi các kết nối tắt (shortcut connections) thường được thiết lập giữa các lớp có số lượng kênh lớn (wide layers), MobileNetV2 áp dụng kết nối tắt giữa các lớp cổ chai mỏng (thin bottleneck layers), từ đó giảm đáng kể chi phí tính toán.

Cụ thể, mỗi khối Inverted Residual bao gồm ba bước chính:



Hình 2.2.1.1. Cấu trúc khối Inverted Residual trong MobileNetV2 (Hẹp => Rộng => Hẹp).

Về mặt toán học, giả sử h_i và w_i lần lượt là chiều cao và chiều rộng của đặc trưng đầu vào. Quá trình biến đổi dữ liệu qua một khối **Inverted Residual** trong MobileNetV2 có thể được mô tả thông qua ba bước chính như sau:

Mở rộng (Expansion):

$$F_{exp} = \text{ReLU6}(\text{Conv}_{1 \times 1}(x, t \cdot k))$$

Trích xuất đặc trưng theo chiều sâu (Depthwise Convolution):

$$F_{depth} = \text{ReLU6}(\text{DConv}_{3 \times 3}(F_{exp}))$$

Nén tuyến tính (Projection):

$$y = \text{Conv}_{1 \times 1}(F_{depth})$$

Trong đó:

- x là tensor đầu vào của khối Inverted Residual.
- k là số kênh của đầu vào.
- t là hệ số mở rộng (Expansion Factor), thường được đặt bằng 6 trong MobileNetV2.
- $\text{Conv}_{1 \times 1}$ là phép tích chập điểm (pointwise convolution).
- $\text{DConv}_{3 \times 3}$ là phép tích chập theo chiều sâu (depthwise convolution).
- $\text{ReLU6}(\cdot)$ là hàm kích hoạt được định nghĩa như sau:

$$\text{ReLU6}(z) = \min(\max(0, z), 6)$$

Hàm kích hoạt ReLU6 giúp giới hạn giá trị đầu ra trong khoảng $[0, 6]$, từ đó phù hợp hơn với các phép tính số học dấu phẩy tĩnh (fixed-point arithmetic), đặc biệt hiệu quả khi triển khai mô hình trên các thiết bị di động và hệ thống nhúng có tài nguyên hạn chế.

Cách tiếp cận này cho phép mô hình duy trì khả năng biểu diễn mạnh mẽ trong khi vẫn đảm bảo hiệu quả về mặt tài nguyên, đồng thời giúp luân chuyển thông tin giữa các lớp một cách hiệu quả hơn.

2.2.2. Linear Bottlenecks

Trong kiến trúc MobileNetV2, lớp cuối cùng của khối Inverted Residual sử dụng hàm kích hoạt tuyến tính (Linear Activation) thay vì hàm phi tuyến ReLU như thông thường. Quyết định thiết kế này dựa trên phân tích lý thuyết về khả năng bảo toàn thông tin của các hàm kích hoạt:

1. Vấn đề của ReLU trong không gian chiều thấp:

Hàm ReLU có tính chất $f(x) = \max(0, x)$. Khi đầu vào có giá trị âm, ReLU sẽ trả về 0. Điều này đồng nghĩa với việc thông tin tại các nơ-ron đó bị "triệt tiêu" hoàn toàn và không thể khôi phục được (non-invertible).

- Trong không gian nhiều chiều (High-dimensional space), việc mất mát này không quá nghiêm trọng vì thông tin có thể được lưu trữ dư thừa ở các kênh khác.
- Tuy nhiên, tại các lớp Bottleneck (nơi số kênh bị nén xuống rất thấp, ví dụ 24 hoặc 32 kênh), việc áp dụng ReLU sẽ khiến một lượng lớn thông tin quan trọng bị xóa sổ vĩnh viễn, làm đứt gãy luồng dữ liệu (Information Collapse).

2. Giải pháp Linear Bottleneck:

Bằng cách loại bỏ hàm kích hoạt phi tuyến tại lớp đầu ra của khối Residual (tức là chỉ thực hiện phép chiếu tuyến tính $y = Wx + b$), MobileNetV2 đảm bảo rằng toàn bộ thông tin đặc trưng sau khi nén được bảo toàn nguyên vẹn để truyền sang khối tiếp theo.

Kết luận: Việc sử dụng Linear Bottleneck là yếu tố then chốt giúp MobileNetV2 duy trì độ chính xác cao ngay cả khi giảm mạnh số lượng tham số và tính toán.

2.2.3. So sánh MobileNetV1 và MobileNetV2

Đặc điểm	MobileNetV1	MobileNetV2
Cấu trúc khối	Depthwise Separable Convolution	Inverted Residual Block
Kết nối tắt (Skip connection)	Không	Có
Hàm kích hoạt	ReLU6	ReLU6 kết hợp Linear Bottleneck
Hiệu suất (trên ImageNet)	Thấp hơn	Cao hơn với cùng số lượng tham số

Bảng 2.2.3.1 So sánh MobileNetV1 và MobileNetV2

2.3. Các công nghệ tích hợp

- **TensorFlow/Keras:** Framework huấn luyện mô hình AI.
- **Flask:** Web Framework nhẹ của Python để xây dựng API.
- **Docker:** Công nghệ Containerization giúp đóng gói ứng dụng để triển khai dễ dàng trên mọi môi trường.
- **MongoDB:** Cơ sở dữ liệu NoSQL lưu trữ lịch sử linh hoạt.
- **OpenWeatherMap API:** Cung cấp dữ liệu thời tiết thời gian thực.

CHƯƠNG 3: THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

3.1. Quy trình xử lý dữ liệu (Data Pipeline)

Trong các bài toán Học sâu (Deep Learning), chất lượng dữ liệu đóng vai trò quyết định đến hiệu suất của mô hình ("Garbage In, Garbage Out"). Để đảm bảo mô hình MobileNetV2 có khả năng học tập tốt và tổng quát hóa cao trên thực tế, hệ thống áp dụng một quy trình xử lý dữ liệu khép kín (End-to-End Data Pipeline) gồm 5 giai đoạn chính, được mô tả chi tiết dưới đây:

3.1.1. Thu thập dữ liệu đa nguồn (Hybrid Data Collection)

Để xây dựng bộ dữ liệu đại diện cho phong cách thời trang đa dạng, đề án áp dụng chiến lược thu thập "Lai" (Hybrid Strategy) kết hợp giữa nguồn dữ liệu học thuật chuẩn và dữ liệu thực tế:

- **Nguồn dữ liệu chuẩn (Benchmark Dataset):** Sử dụng tập dữ liệu nền tảng từ **Kaggle Fashion Dataset** và một phần của **DeepFashion**. Ưu điểm của nguồn này là ảnh có độ phân giải cao, phông nền sạch và nhãn được gán chính xác. Tuy nhiên, hạn chế là các mẫu trang phục đôi khi quá "lý tưởng" hoặc mang tính biểu diễn, thiếu tính thực tế của ảnh chụp đời thường.
- **Nguồn dữ liệu thực tế (Real-world Data):** Để khắc phục hạn chế trên, đề án sử dụng các **Script Python** (sử dụng thư viện BeautifulSoup và Selenium) để cào dữ liệu (Crawling) tự động từ các công cụ tìm kiếm hình ảnh (Bing Images, Google Images) và các trang thương mại điện tử phổ biến tại Việt Nam. Các từ khóa tìm kiếm (Keywords) được tối ưu hóa cho 20 loại trang phục mục tiêu (Ví dụ: "Áo polo nam", "Váy công sở", "Vintage hoodie" ...).
- **Kết quả:** Quy trình này giúp thu thập được lượng lớn dữ liệu thô, bao gồm cả những ảnh có điều kiện ánh sáng kém, góc chụp nghiêng hoặc phông nền phức tạp, giúp mô hình thích nghi tốt hơn với môi trường sử dụng thực tế của người dùng.

3.1.2. Tiền xử lý và Làm sạch dữ liệu (Data Cleaning)

Dữ liệu thô sau khi thu thập thường chứa nhiều nhiễu. Quy trình làm sạch được thực hiện tự động kết hợp bán tự động qua các bước:

1. **Kiểm tra định dạng:** Sử dụng thư viện PIL (Python Imaging Library) để quét toàn bộ thư mục, loại bỏ các tệp tin bị lỗi header (corrupted files) hoặc không phải định dạng ảnh tiêu chuẩn (.jpg, .png, .jpeg).
2. **Khử trùng lặp (De-duplication):** Khi thu thập từ internet, việc trùng lặp ảnh là không tránh khỏi. Hệ thống sử dụng giải thuật **Mã băm MD5 (Message-Digest Algorithm 5)** để tạo ra một chuỗi ký tự duy nhất (checksum) cho mỗi bức ảnh. Các ảnh có cùng mã băm sẽ bị xóa bỏ, đảm bảo tính duy nhất của dữ liệu huấn luyện.
3. **Lọc nhiễu thủ công:** Thực hiện rà soát ngẫu nhiên để loại bỏ các ảnh không chứa đối tượng quần áo (ví dụ: ảnh móc treo, ảnh tủ gỗ) hoặc ảnh chứa quá nhiều vật thể gây nhiễu, đảm bảo nhãn dữ liệu chính xác tuyệt đối.

3.1.3. Phân chia tập dữ liệu (Data Splitting)

Bộ dữ liệu sạch được chia thành 3 tập con độc lập theo tỷ lệ vàng **80:10:10** để phục vụ quá trình huấn luyện và kiểm thử khách quan:

- **Tập huấn luyện (Training Set - 80%):** Dùng để cập nhật trọng số của mạng nơ-ron thông qua thuật toán lan truyền ngược (Backpropagation).
- **Tập kiểm định (Validation Set - 10%):** Dùng để tinh chỉnh các siêu tham số (Hyperparameters) và đánh giá mô hình sau mỗi Epoch nhằm phát hiện sớm hiện tượng Overfitting.
- **Tập kiểm thử (Test Set - 10%):** Được giữ hoàn toàn bí mật trong quá trình huấn luyện, chỉ dùng một lần duy nhất để đánh giá hiệu năng thực tế cuối cùng của hệ thống.

3.1.4. Tăng cường dữ liệu (Data Augmentation)

Do số lượng mẫu dữ liệu thực tế có hạn, kỹ thuật Tăng cường dữ liệu được áp dụng để nhân rộng số lượng mẫu và giúp mô hình học được các đặc trưng bất biến (Invariant Features). Đồ án sử dụng lớp ImageDataGenerator của Keras để thực hiện các biến đổi ngẫu nhiên theo thời gian thực (On-the-fly Augmentation):

- **Lật ngang (Horizontal Flip):** Mô phỏng tính đối xứng của quần áo.
- **Xoay (Rotation):** Xoay ảnh ngẫu nhiên trong khoảng $\pm 15^\circ$ để mô phỏng các góc chụp nghiêng của người dùng.
- **Dịch chuyển (Shift):** Dịch chuyển khung hình theo chiều dọc/ngang (khoảng 10%) để mô hình học cách nhận diện vật thể ngay cả khi không nằm chính giữa khung hình.
- **Zoom & Shear:** Phóng to và biến dạng nhẹ để tăng tính đa dạng về tỷ lệ.

3.1.5. Chuẩn hóa đầu vào (Normalization)

Trước khi đưa vào mạng MobileNetV2, dữ liệu cần được chuẩn hóa để phù hợp với kiến trúc mạng và tăng tốc độ hội tụ:

1. **Resize:** Tất cả ảnh đầu vào (bất kể kích thước gốc) đều được đưa về kích thước cố định **224 x 224 pixels** sử dụng phương pháp nội suy (Interpolation) để giữ chi tiết tốt nhất.
2. **Rescaling:** Giá trị pixel của ảnh (nằm trong khoảng $[0, 255]$) được chia cho 255 để đưa về khoảng giá trị thực **$[0, 1]$** . Việc đưa dữ liệu về khoảng nhỏ giúp thuật toán tối ưu (Adam) hoạt động hiệu quả hơn, tránh hiện tượng bão hòa gradient (gradient saturation) ở các lớp kích hoạt.

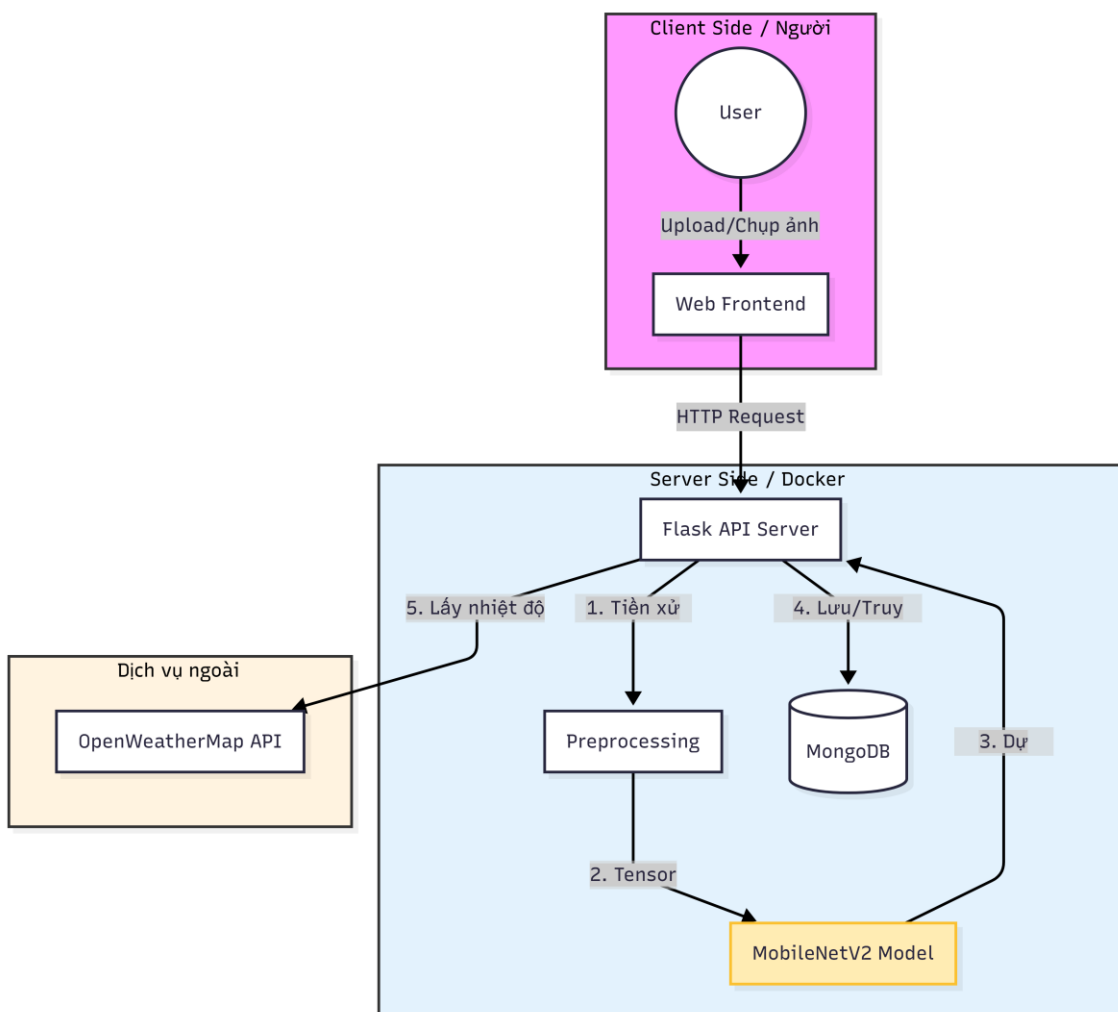
3.2. Kiến trúc hệ thống

Hệ thống được thiết kế theo mô hình **Client-Server** hiện đại, đảm bảo khả năng mở rộng và bảo trì dễ dàng. Kiến trúc tổng thể bao gồm 3 phân hệ chính:

- **Frontend (Client Side):** Giao diện Web tương tác với người dùng, cung cấp các chức năng tải ảnh (upload) hoặc chụp ảnh trực tiếp (capture).

- **Backend (Server Side):** Xây dựng trên nền tảng Flask API, chịu trách nhiệm nhận request, thực hiện tiền xử lý ảnh, gọi mô hình AI (MobileNetV2) để dự đoán và trả về kết quả dưới dạng JSON.
- **Database & External Services:** Sử dụng MongoDB để lưu trữ lịch sử và tích hợp OpenWeatherMap API để lấy dữ liệu thời tiết thời gian thực.

Sơ đồ dưới đây minh họa sự tương tác giữa các thành phần:



Hình 3.2.1 Sơ đồ kiến trúc tổng thể của hệ thống.

3.3. Thiết kế Cơ sở dữ liệu (Database Schema)

Hệ thống quyết định sử dụng hệ quản trị cơ sở dữ liệu **MongoDB** (thuộc nhóm NoSQL - Document Store) làm giải pháp lưu trữ chính cho phân hệ lịch sử người dùng. Quyết định thay thế các giải pháp SQL truyền thống (như MySQL, PostgreSQL) được dựa trên những phân tích kỹ thuật sau:

- **Sự tương thích dữ liệu tự nhiên (Native Data Compatibility):** Kết quả trả về từ mô hình AI và metadata của hình ảnh bản chất là dữ liệu bán cấu trúc (Semi-structured). Định dạng lưu trữ BSON (Binary JSON) của MongoDB tương thích hoàn toàn với cấu trúc Dictionary/JSON của Python và JavaScript. Điều này giúp loại bỏ sự phức tạp của quá trình chuyển đổi dữ liệu (Object-Relational Mapping - ORM) và đơn giản hóa logic lập trình.
- **Tối ưu hóa hiệu năng (High Performance I/O):** Với đặc thù của chức năng "Lịch sử quét", hệ thống yêu cầu tốc độ ghi (Write-intensive) cao để lưu vết log ngay lập tức sau khi AI xử lý xong, đồng thời cần tốc độ truy vấn (Read) nhanh để hiển thị danh sách trên giao diện người dùng. MongoDB cung cấp khả năng xử lý thông lượng lớn với độ trễ thấp (Low latency) nhờ cơ chế không ràng buộc quan hệ phức tạp (Joins).
- **Tính linh hoạt trong phát triển (Agile Scalability):** Trong quá trình phát triển và nâng cấp mô hình AI, các trường dữ liệu mới (ví dụ: thêm thông tin thời tiết, tọa độ GPS, phản hồi người dùng) có thể phát sinh liên tục. MongoDB hỗ trợ lược đồ động (Dynamic Schema), cho phép thay đổi cấu trúc dữ liệu mà không cần thực hiện các thao tác di trú (Migration) nặng nề hay làm gián đoạn dịch vụ như trong SQL truyền thống.

3.3.1. Cấu trúc Collection "History"

Dữ liệu lịch sử quét của người dùng được lưu trữ trong 3 wardrobe_db,

Collection history với lược đồ (Schema) chi tiết như sau:

Tên trường (Field)	Kiểu dữ liệu	Mô tả chi tiết
_id	ObjectId	Khóa chính duy nhất (Primary Key) được MongoDB tự động sinh ra cho mỗi bản ghi.
filename	String	Tên file ảnh hệ thống, được mã hóa bằng thuật toán UUID (Ví dụ: 550e8400-e29b...jpg) để đảm bảo tính duy nhất, tránh ghi đè khi người dùng upload ảnh trùng tên.
original_name	String	Tên file gốc do người dùng tải lên (Ví dụ: anh_ao_thun.jpg), phục vụ việc hiển thị thân thiện với người dùng.
label	String	Nhãn dự đoán từ mô hình MobileNetV2 (Ví dụ: Ao_hoodie, Quan_short...).
confidence	Double	Độ tin cậy của dự đoán, lưu dưới dạng số thực (0.0 - 100.0). Hệ thống chỉ lưu các kết quả có độ tin cậy $\geq 81.0\%$.
timestamp	DateTime	Thời gian thực hiện phân loại, sử dụng múi giờ hệ thống server. Dùng để sắp xếp hiển thị dòng thời gian.
recommendation	String	Chuỗi văn bản chứa lời khuyên phối đồ (Styling advice) tương ứng với nhãn được dự đoán.

Bảng 3.3.1.1. Đặc tả trường dữ liệu (Field Specification)

3.3.2. Mẫu dữ liệu (Data Sample)

Dưới đây là biểu diễn JSON của một bản ghi thực tế được lưu trữ trong hệ thống:

```
JSON
{
  "_id": { "$oid": "65b3f2a1c92b4a0012345678" },
  "filename": "f47ac10b-58cc-4372-a567-0e02b2c3d479.jpg",
  "original_name": "IMG_2026.jpg",
  "label": "Ao_len",
  "confidence": 98.5,
  "timestamp": { "$date": "2026-01-28T22:30:00.000Z" },
  "recommendation": "Giu am va thoi trang. Mac ngoai so mi trang de lo co ao la cach phoi kinh dien (Layering)."
}
```

3.3.3. Cơ chế lưu trữ vật lý & An toàn dữ liệu

Do hệ thống được triển khai trên nền tảng **Docker Container**, vấn đề an toàn dữ liệu (Data Persistence) được giải quyết thông qua cơ chế **Bind Mount**:

- **Docker Volume Mapping:** Hệ thống ánh xạ thư mục thực tế `./db_data` trên máy chủ (Host) vào thư mục `/data/db` bên trong Container MongoDB.
- **Cấu hình Docker Compose:**

YAML

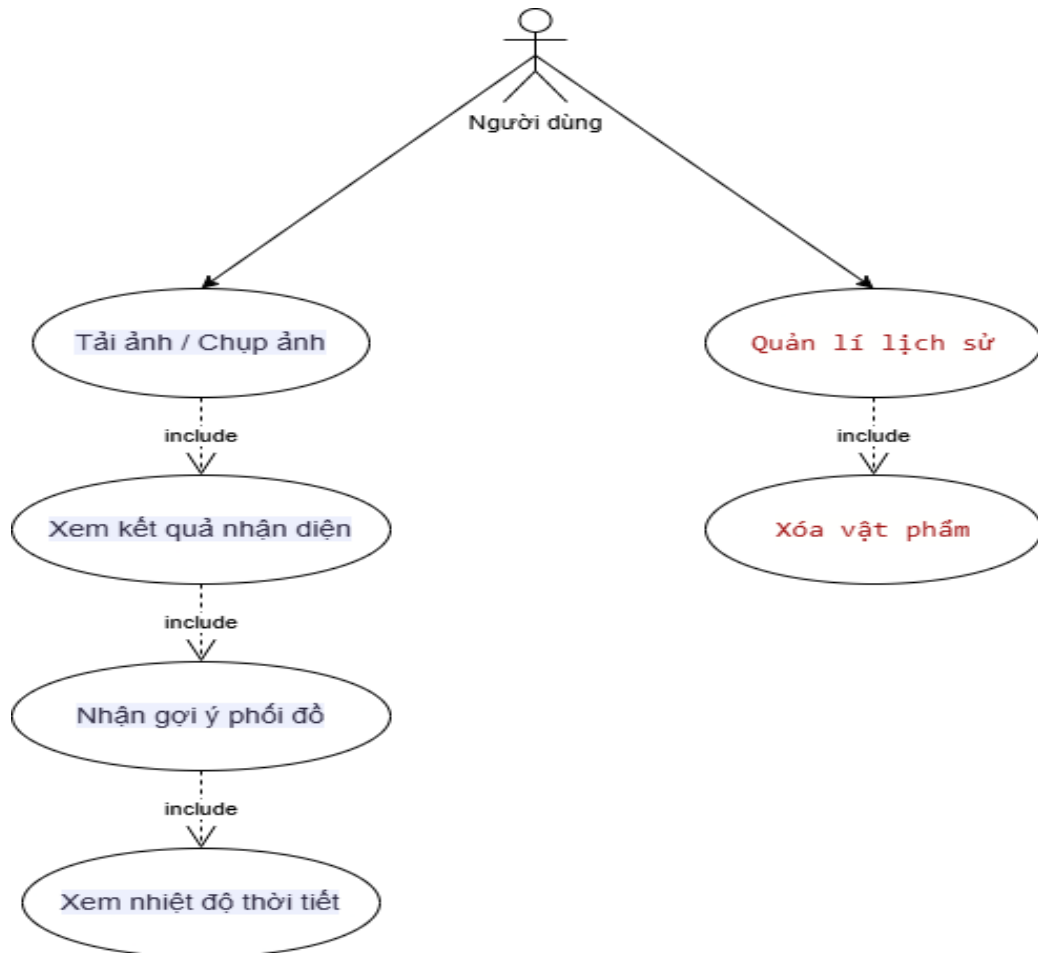
volumes:

```
- ./db_data:/data/db
```

- **Ưu điểm:** Cơ chế này đảm bảo dữ liệu lịch sử và hình ảnh không bị mất đi ngay cả khi Container bị xóa hoặc khởi động lại, đồng thời cho phép người quản trị dễ dàng sao lưu (Backup) bằng cách copy thư mục vật lý.

3.4. Phân tích chức năng hệ thống (Use-case Description)

Hệ thống bao gồm các chức năng nghiệp vụ chính phục vụ người dùng cuối, được mô tả qua biểu đồ Use-case dưới đây:



Hình 3.4.1 Biểu đồ ca sử dụng (Use-case) của hệ thống.

Hệ thống được thiết kế xoay quanh trải nghiệm người dùng liền mạch, kết hợp sức mạnh của trí tuệ nhân tạo với dữ liệu thời gian thực. Các chức năng nghiệp vụ chính được mô tả chi tiết như sau:

1. Thu thập và Kiểm tra dữ liệu ảnh (Image Acquisition & Validation)

Đây là công giao tiếp đầu tiên giữa người dùng và hệ thống. Chức năng này không chỉ đơn thuần là tải ảnh mà còn bao gồm các bước tiền xử lý nghiêm ngặt:

- **Giao diện đa phương thức:** Người dùng có thể tải ảnh lên từ thư viện thiết bị hoặc kích hoạt camera để chụp ảnh trực tiếp thông qua giao diện Web Responsive.
- **Cơ chế xác thực an toàn (Security Validation):** Trước khi ảnh được gửi về Server, hệ thống thực hiện kiểm tra kép tại Client. Về mặt định dạng, chỉ chấp nhận các tệp ảnh tiêu chuẩn (.jpg, .png). Về mặt dung lượng, hệ thống giới hạn kích thước tệp dưới 5MB để tối ưu hóa băng thông và ngăn chặn nguy cơ tấn công từ chối dịch vụ (DoS).
- **Hiển thị xem trước (Client-side Preview):** Ảnh sau khi chọn sẽ được hiển thị ngay lập tức trên trình duyệt (Preview) giúp người dùng xác nhận đúng đối tượng trước khi nhấn nút xử lý.

2. Phân loại trang phục cốt lõi (Core AI Classification)

Đây là chức năng trung tâm, nơi mô hình Deep Learning thực hiện nhiệm vụ nhận thức thị giác máy tính:

- **Tiền xử lý ảnh (Preprocessing):** Ảnh đầu vào được tự động thay đổi kích thước (Resize) về chuẩn 224x224 pixel và chuẩn hóa giá trị điểm ảnh (Normalization) về khoảng [0, 1] để phù hợp với yêu cầu đầu vào của mạng nơ-ron.
- **Suy luận thời gian thực (Real-time Inference):** Mô hình MobileNetV2 thực hiện trích xuất đặc trưng và tính toán xác suất cho các nhóm nhãn (Classes) khác nhau như: *Ao_hoodie*, *Quan_short*, *Vay*, *Giay_sneaker*...
- **Ngưỡng tin cậy (Confidence Threshold):** Để đảm bảo độ chính xác, hệ thống thiết lập ngưỡng tin cậy tối thiểu là **81%**. Các kết quả dự đoán dưới ngưỡng này sẽ được cảnh báo là "Không xác định", giúp loại bỏ các kết quả nhiễu.

3. Hệ thống tư vấn theo ngữ cảnh (Contextual Recommendation Engine)

Điểm khác biệt của ứng dụng là khả năng kết hợp kết quả phân loại tĩnh với dữ liệu môi trường động:

- **Tích hợp dữ liệu thời tiết:** Hệ thống kết nối API bên thứ ba (OpenWeatherMap) để lấy dữ liệu nhiệt độ thời gian thực tại vị trí người dùng (Đà Nẵng).
- **Logic gợi ý thông minh (Rule-based Logic):** Dựa trên nhãn trang phục vừa nhận diện và nhiệt độ hiện tại, hệ thống tra cứu tập luật để đưa ra lời khuyên.
 - *Ví dụ:* Nếu nhận diện là "Ao_hoodie" và nhiệt độ < 20°C, hệ thống gợi ý: "*Trời lạnh, rất hợp để mặc Hoodie giữ ấm*".
 - *Ví dụ:* Nếu nhận diện là "Quan_short" nhưng nhiệt độ > 30°C, hệ thống gợi ý: "*Thời tiết nóng, quần short là lựa chọn hoàn hảo*".

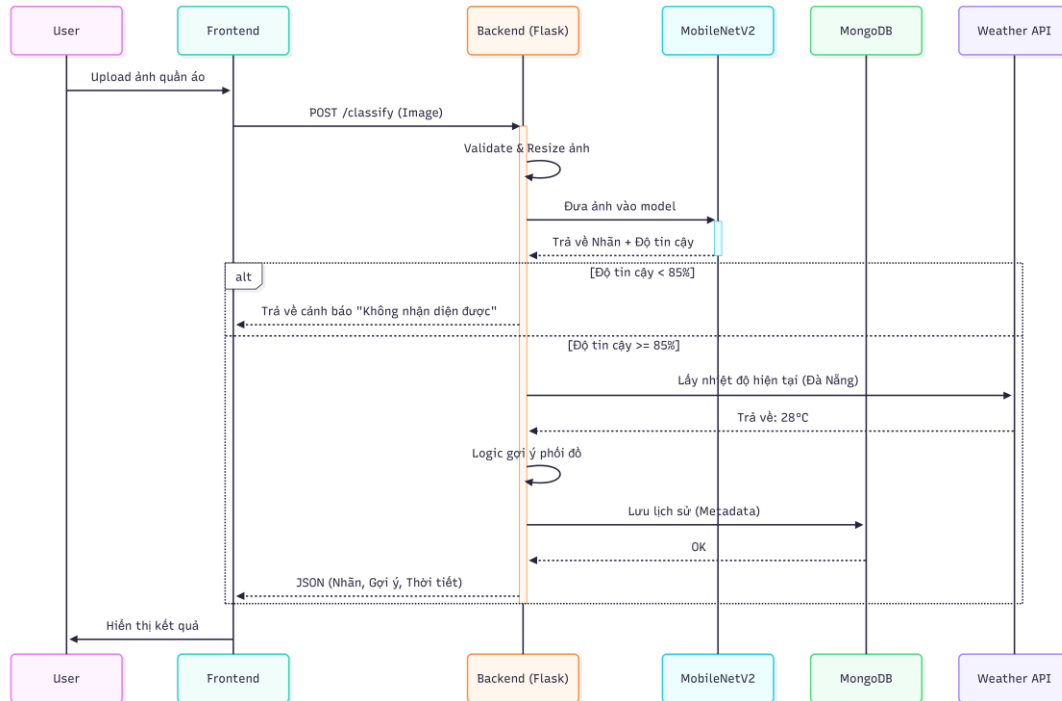
4. Quản lý Tủ đồ số hóa (Digital Wardrobe Management)

Chức năng này giúp người dùng số hóa và quản lý thói quen ăn mặc cá nhân:

- **Lưu trữ bền vững:** Toàn bộ lịch sử quét bao gồm ảnh, nhãn tên, độ tin cậy và thời gian đều được lưu trữ vĩnh viễn trong cơ sở dữ liệu MongoDB.
- **Bộ lọc thông minh (Smart Filtering):** Người dùng có thể lọc danh sách lịch sử theo các nhóm danh mục lớn như: *Áo, Quần, Giày Dép, Phụ kiện*. Chức năng này giúp việc tìm kiếm lại các món đồ cũ trở nên nhanh chóng thay vì phải lướt qua toàn bộ danh sách.
- **Thao tác quản trị:** Hỗ trợ người dùng chủ động xóa các mục lịch sử sai hoặc không còn cần thiết.

3.5. Thiết kế luồng xử lý chi tiết (Sequence Diagram)

Để làm rõ logic nghiệp vụ, đặc biệt là cơ chế kiểm soát ngưỡng tin cậy (Threshold) và gợi ý thông minh, đồ án xây dựng biểu đồ tuần tự cho luồng xử lý chính:



Hình 3.5.1 Biểu đồ tuần tự xử lý luồng nhận diện và gợi ý.

Quy trình xử lý:

1. Frontend gửi ảnh lên API.
2. Backend kiểm tra tính hợp lệ và tiền xử lý ảnh.
3. Model trả về kết quả dự đoán. Nếu độ tin cậy < **81%**, hệ thống trả về cảnh báo "Không nhận diện được".
4. Nếu độ tin cậy \geq **81%**, hệ thống tiếp tục lấy thông tin thời tiết và truy vấn cơ sở dữ liệu để tạo gợi ý phù hợp nhất.

3.6. Các giải pháp bảo mật và an toàn hệ thống

Trong bối cảnh các mối đe dọa an ninh mạng ngày càng gia tăng, việc đảm bảo tính toàn vẹn, bí mật và sẵn sàng cho hệ thống là ưu tiên hàng đầu. Đồ án đã áp dụng mô hình bảo mật nhiều lớp (Defense in Depth), kết hợp giữa kiểm soát mã nguồn ứng dụng và cô lập hạ tầng triển khai.

3.6.1. Kiểm tra và xác thực dữ liệu đầu vào (Strict Input Validation)

Hệ thống coi mọi dữ liệu từ phía người dùng là "không tin cậy" (Untrusted Input) và áp dụng quy trình lọc nghiêm ngặt:

- **Xác thực định dạng tệp (File Type Validation):** Không chỉ dựa vào phần mở rộng (extension) dễ bị giả mạo, hệ thống sử dụng thư viện xử lý ảnh chuyên dụng (Pillow/PIL) để phân tích cấu trúc nhị phân của tệp (Magic Bytes). Điều này ngăn chặn tuyệt đối các cuộc tấn công **RCE (Remote Code Execution)** bằng cách ngụy trang các tệp mã độc (như .php, .sh, .exe) dưới vỏ bọc hình ảnh.
- **Giới hạn dung lượng (Size Limiting):** Cấu hình giới hạn kích thước tệp tải lên (ví dụ: tối đa 5MB) được thiết lập ngay tại tầng ứng dụng Flask. Cơ chế này giúp giảm thiểu nguy cơ tấn công từ chối dịch vụ (DoS), ngăn chặn việc kẻ tấn công làm tràn bộ nhớ đệm hoặc gây tắc nghẽn băng thông máy chủ bằng các tệp tin khổng lồ.

3.6.2. Cơ chế chia sẻ tài nguyên nguồn gốc chéo (CORS Policies)

Để bảo vệ API Backend khỏi các truy cập trái phép từ các trình duyệt hoặc tên miền lạ, hệ thống triển khai cấu hình **CORS (Cross-Origin Resource Sharing)**:

- Hệ thống kiểm soát các HTTP Header như Access-Control-Allow-Origin, chỉ cho phép các yêu cầu xuất phát từ các định danh tin cậy (trong môi trường phát triển là Frontend ReactJS).

- Các yêu cầu phức tạp (như POST, DELETE) buộc phải thông qua bước kiểm tra sơ bộ (**Preflight Request - OPTIONS**) để đảm bảo máy chủ chấp nhận phương thức và tiêu đề trước khi thực sự xử lý dữ liệu.

3.6.3. Làm sạch dữ liệu và Chống tấn công đường dẫn (Anti-Path Traversal)

Lỗ hổng Path Traversal (hay Directory Traversal) cho phép kẻ tấn công truy cập vào các thư mục hệ thống bằng các ký tự như ../. Để khắc phục:

- **Đổi tên tệp ngẫu nhiên (UUID naming):** Hệ thống không bao giờ sử dụng tên gốc của tệp tin do người dùng tải lên. Thay vào đó, mỗi tệp được gán một định danh duy nhất (UUIDv4).
- **Lợi ích kép:** Phương pháp này vừa loại bỏ hoàn toàn khả năng thực thi các ký tự đặc biệt trong tên file, vừa ngăn chặn tình trạng **ghi đè tệp (File Overwriting)** hoặc tấn công liệt kê tài nguyên (Resource Enumeration) do đoán được tên file theo thứ tự.

3.6.4. Cô lập môi trường bằng Docker (Container Isolation)

Tận dụng kiến trúc ảo hóa ở cấp độ hệ điều hành (Containerization) của Docker để tạo ra các "hộp cát" (Sandbox) an toàn:

- **Mạng nội bộ (Internal Network):** Database MongoDB không được public trực tiếp ra Internet. Nó hoạt động trong một mạng ảo riêng (wardrobe_network) và chỉ chấp nhận kết nối từ Backend Container thông qua tên định danh nội bộ. Điều này giảm thiểu tối đa bề mặt tấn công (Attack Surface) vào cơ sở dữ liệu.
- **Môi trường độc lập:** Các thư viện và tiến trình của ứng dụng chạy trong môi trường Linux Alpine tối giản, tách biệt hoàn toàn với hệ điều hành của máy chủ vật lý, hạn chế rủi ro leo thang đặc quyền (Privilege Escalation) nếu ứng dụng bị xâm nhập.

3.6.5. Quản lý cấu hình và Bí mật (Secrets Management)

Hệ thống tuân thủ nghiêm ngặt nguyên tắc thứ III (Config) trong bộ tiêu chuẩn **The Twelve-Factor App** – một phương pháp luận tiêu chuẩn vàng trong xây dựng ứng dụng phần mềm hiện đại (SaaS). Theo đó, hệ thống thực hiện tách biệt hoàn toàn **Cấu hình (Config)** ra khỏi **Mã nguồn (Code)**.

a. Loại bỏ thông tin nhạy cảm khỏi mã nguồn (Hard-coding Elimination)

Các thông tin nhạy cảm và thay đổi theo môi trường triển khai như:

- Chuỗi kết nối cơ sở dữ liệu (MONGO_URI).
- Khóa xác thực dịch vụ bên thứ ba (OpenWeatherMap API Key).
- Các cờ gỡ lỗi (Debug flags).

Tuyệt đối không được ghi cứng (Hard-code) trực tiếp trong các file mã nguồn (.py, .js). Việc ghi cứng các thông tin này tiềm ẩn rủi ro bảo mật nghiêm trọng: nếu mã nguồn bị lộ (ví dụ: vô tình push lên public repository như GitHub), toàn bộ thông tin đăng nhập hệ thống sẽ bị kẻ tấn công khai thác ngay lập tức. Hơn nữa, việc thay đổi cấu hình sẽ yêu cầu phải sửa code và build lại toàn bộ ứng dụng, gây gián đoạn dịch vụ.

b. Cơ chế nạp qua Biến môi trường (Environment Variables Injection) Thay vào đó, đồ án áp dụng cơ chế nạp cấu hình động thông qua **Biến môi trường (Environment Variables)** tại thời điểm Runtime (khi ứng dụng khởi chạy). Quy trình này được thực hiện thông qua sự phối hợp giữa Docker và ứng dụng Flask:

1. **Tại tầng Triển khai (Infrastructure Layer):** Trong file cấu hình docker-compose.yml, các giá trị cấu hình được định nghĩa dưới dạng các cặp khóa-giá trị (Key-Value) trong mục environment. Docker sẽ chịu trách nhiệm tiêm (inject) các giá trị này vào không gian bộ nhớ của Container khi khởi tạo.

2. **Tại tầng Ứng dụng (Application Layer):** Mã nguồn Python sử dụng thư viện chuẩn `os` để truy xuất các giá trị này thông qua hàm `os.environ.get('KEY_NAME')`. Nếu biến môi trường không tồn tại, hệ thống sẽ tự động sử dụng giá trị mặc định an toàn hoặc báo lỗi khởi động, đảm bảo tính nhất quán của hệ thống.

c. Lợi ích về tính linh hoạt và bảo mật (Flexibility & Security) Cách tiếp cận này mang lại hai lợi ích cốt lõi cho hệ thống:

- **Bảo mật mã nguồn:** Mã nguồn trở nên "sạch" (Clean Code). Bất kỳ ai có quyền truy cập mã nguồn cũng không thể biết được mật khẩu DB hay API Key thực tế, vì chúng chỉ tồn tại trong môi trường server nơi ứng dụng chạy.
- **Linh hoạt trong triển khai (Deployment Portability):** Cùng một Docker Image có thể chạy trên nhiều môi trường khác nhau (Local, Testing, Production) chỉ bằng cách thay đổi giá trị biến môi trường mà không cần thay đổi bất kỳ dòng code nào. Ví dụ: Ở môi trường Local dùng DB nội bộ, nhưng khi lên Production chỉ cần đổi biến `MONGO_URI` để trỏ tới Cloud Cluster.

CHƯƠNG 4: THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ (QUAN TRỌNG NHẤT)

4.1. Xây dựng và Xử lý Bộ dữ liệu (Dataset Construction)

4.1.1. Chiến lược Dữ liệu lai (Hybrid Dataset)

Để giải quyết thách thức về sự đa dạng của trang phục trong môi trường thực tế, đề án không chỉ dựa vào một nguồn dữ liệu duy nhất mà áp dụng chiến lược **Hybrid Dataset** kết hợp từ ba nguồn chính:

1. **Nguồn dữ liệu chuẩn (Benchmark Datasets):** Sử dụng hình ảnh từ các bộ dữ liệu học thuật uy tín như *DeepFashion* và *Kaggle Fashion Product Images* để làm cơ sở (Baseline). Đặc điểm của nhóm này là ảnh có độ phân giải cao, phong nền sạch, giúp mô hình học các đặc trưng cơ bản tốt.
2. **Thu thập dữ liệu thực tế (Web Crawling):** Đề mô hình có khả năng thích ứng với ảnh chụp đời thường (street style, flat-lay, ảnh selfie), nhóm đã phát triển các công cụ thu thập dữ liệu tự động từ *Bing Images* và các sàn thương mại điện tử.
 - *Xử lý kỹ thuật:* Hệ thống crawler được tích hợp cơ chế vượt qua các rào cản bảo mật (lỗi SSL, 403 Forbidden) và tự động lọc ảnh lỗi (corrupted images).

4.1.2. Quy trình Tiền xử lý (Preprocessing)

- **Lọc trùng lặp:** Sử dụng thuật toán mã băm (Hashing MD5) để loại bỏ các ảnh trùng lặp, đảm bảo tính duy nhất cho dữ liệu huấn luyện.
- **Tăng cường dữ liệu (Data Augmentation):** Áp dụng kỹ thuật *Offline Augmentation* (nhân bản dữ liệu trước khi train) bao gồm các thao tác: lật ngang (horizontal flip), điều chỉnh độ sáng/trương phản. Điều này giúp giải quyết bài toán mất cân bằng dữ liệu ở các lớp thiểu số.



Hình 4.1.2.1 Ảnh trước và sau khi biến đổi

Kỹ thuật	Tham số cấu hình	Mục đích
Rotation	Range: 15 độ	Mô phỏng việc người dùng chụp ảnh sản phẩm bị nghiêng.
Width/Height Shift	Range: 0.1 (10%)	Mô phỏng vật thể không nằm chính giữa khung hình.
Shear / Zoom	Range: 0.1	Mô phỏng biến dạng phối cảnh khi chụp góc xéo.
Horizontal Flip	True	Tăng gấp đôi dữ liệu cho các vật thể đối xứng (như áo, quần).
Fill Mode	Nearest	Xử lý các điểm ảnh bị khuyết sau khi xoay/dịch chuyển.

Bảng 4.1.2.1 Các kỹ thuật tăng cường dữ liệu (Data Augmentation) áp dụng

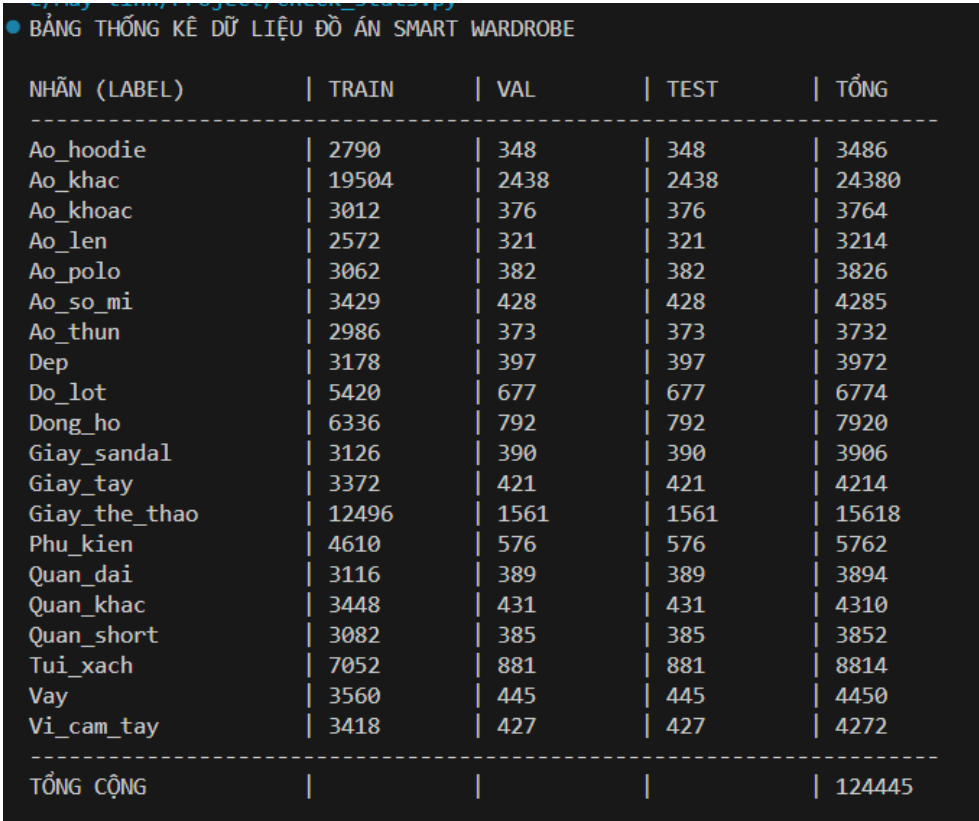
Lý do lựa chọn Offline Augmentation:

Thay vì sử dụng *Online Augmentation* (biến đổi ngay trong quá trình train), đồ án lựa chọn **Offline Augmentation** (sinh dữ liệu trước và lưu vào đĩa) vì 2 lý do chính:

- **Cân bằng dữ liệu (Class Balancing):** Các lớp thiếu số như Ao_hoodie (chỉ có ~3000 ảnh) được nhân bản lên nhiều lần để cân bằng với các lớp đa số như Ao_khac (~24000 ảnh), giúp model không bị thiên kiến (Bias).
- **Tốc độ huấn luyện:** Giảm tải cho CPU trong quá trình training, giúp tận dụng tối đa sức mạnh GPU cho việc tính toán mạng nơ-ron.

4.1.3. Thống kê dữ liệu

Sau quá trình làm sạch và xử lý, bộ dữ liệu cuối cùng có các thông số sau:



```
● BẢNG THỐNG KÊ DỮ LIỆU ĐỒ ÁN SMART WARDROBE
```

NHÃN (LABEL)	TRAIN	VAL	TEST	TỔNG
Ao_hoodie	2790	348	348	3486
Ao_khac	19504	2438	2438	24380
Ao_khoac	3012	376	376	3764
Ao_len	2572	321	321	3214
Ao_polo	3062	382	382	3826
Ao_so_mi	3429	428	428	4285
Ao_thun	2986	373	373	3732
Dep	3178	397	397	3972
Do_lot	5420	677	677	6774
Dong_ho	6336	792	792	7920
Giay_sandal	3126	390	390	3906
Giay_tay	3372	421	421	4214
Giay_the_thao	12496	1561	1561	15618
Phu_kien	4610	576	576	5762
Quan_dai	3116	389	389	3894
Quan_khac	3448	431	431	4310
Quan_short	3082	385	385	3852
Tui_xach	7052	881	881	8814
Vay	3560	445	445	4450
Vi_cam_tay	3418	427	427	4272
TỔNG CỘNG				124445

Hình 4.1.3.1 Bảng Thống Kê Dữ Liệu

Sau quá trình làm sạch, lọc trùng lặp và phân chia, bộ dữ liệu cuối cùng có các thông số chi tiết như sau:

- **Tổng quy mô: 124.445 hình ảnh.**
- **Số lượng danh mục (Classes): 20 lớp**, bao gồm các loại trang phục (Ao_polo, Ao_hoodie...) và phụ kiện (Dong_ho, Tui_xach...).
- **Phân bố dữ liệu:** Dữ liệu được chia ngẫu nhiên theo tỷ lệ chiến lược **80% - 10% - 10%**:
 - **Training Set (80%): 99.556 ảnh** (Dùng để huấn luyện trọng số mô hình).
 - **Validation Set (10%): 12.444 ảnh** (Dùng để đánh giá và tinh chỉnh trong quá trình huấn luyện).
 - **Test Set (10%): 12.445 ảnh** (Dữ liệu độc lập hoàn toàn dùng để đánh giá hiệu năng cuối cùng).

4.2. Thực nghiệm Lựa chọn Mô hình (Model Benchmarking)

Để lựa chọn kiến trúc tối ưu cho bài toán triển khai trên thiết bị di động và Web (yêu cầu phản hồi thời gian thực), đề án đã thực hiện so sánh thực nghiệm giữa 3 kiến trúc CNN phổ biến: **MobileNetV2**, **EfficientNetB0**, và **ResNet50**.

4.2.1. Thiết lập thực nghiệm

Các mô hình được huấn luyện trong cùng điều kiện môi trường để đảm bảo tính công bằng:

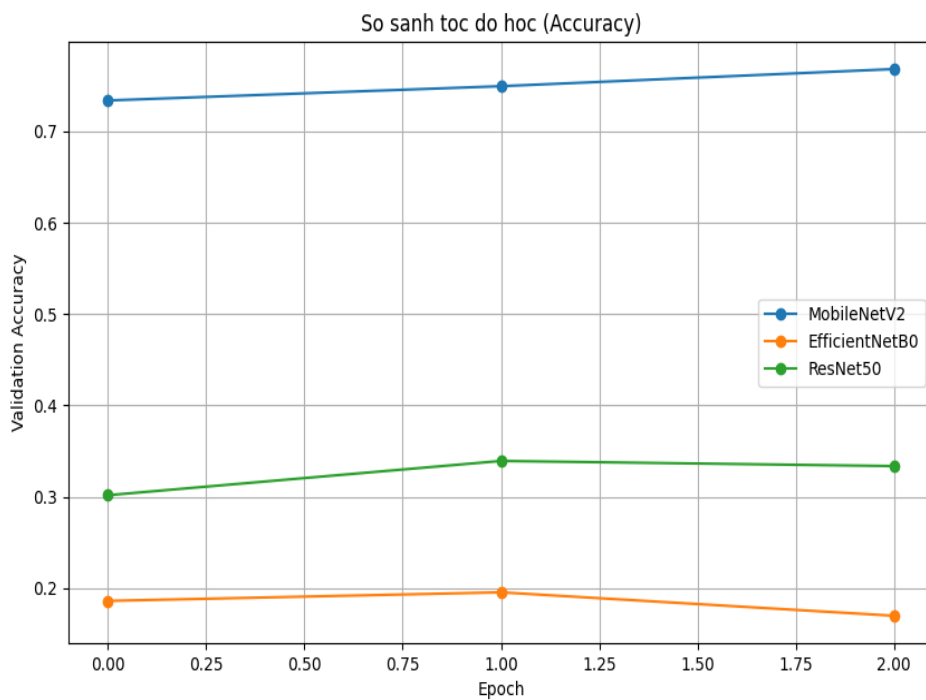
- **Input Size:** 224x224 pixels.
- **Batch Size:** 32.
- **Epochs (Sprint Run):** 3 (để đánh giá tốc độ hội tụ ban đầu).
- **Môi trường:** TensorFlow/Keras trên GPU hỗ trợ.

4.2.2. Kết quả so sánh

Kết quả thực nghiệm được ghi nhận như sau:

Mô hình	Độ chính xác (Val Accuracy)	Độ trễ trung bình (Latency)	Thời gian huấn luyện (3 Epochs)
MobileNetV2	76.81% (Cao nhất)	21.95 ms (Nhanh nhất)	716s
EfficientNetB0	17.00%	38.73 ms	727s
ResNet50	33.38%	72.04 ms	1588s

Bảng 4.2.2.1 So sánh độ hiệu quả của các mô hình



Hình 4.2.2.1 Biểu đồ so sánh Accuracy qua các Epochs đầu tiên của 3 mô hình.

4.2.3. Phân tích và Lựa chọn mô hình tối ưu cho đề án

Nhìn vào biểu đồ trên, ta có thể thấy sự phân hóa rõ rệt về hiệu năng:

- **MobileNetV2 (Đường màu xanh dương):** Thể hiện sự vượt trội hoàn toàn. Ngay từ Epoch 0, độ chính xác đã đạt trên **70%** và duy trì đà tăng ổn định, tiệm cận mức **77%** ở Epoch 2. Điều này chứng tỏ kiến trúc *Inverted Residuals* cực kỳ phù hợp để trích xuất đặc trưng từ bộ dữ liệu thời trang này.
- **ResNet50 (Đường màu xanh lá):** Khởi đầu khá chậm chạp với độ chính xác chỉ đạt khoảng 30-33%, cho thấy mô hình cần nhiều dữ liệu và thời gian hơn để tinh chỉnh tham số.
- **EfficientNetB0 (Đường màu cam):** Cho kết quả thấp nhất (~17-19%), có thể do kiến trúc *Compound Scaling* đòi hỏi chiến lược tiền xử lý phức tạp hơn.

Kết luận: Dựa trên cả hai yếu tố **Tốc độ hội tụ nhanh nhất** và **Độ trễ thấp nhất**, đề án quyết định lựa chọn **MobileNetV2** làm mô hình chính thức để huấn luyện chuyên sâu (Fine-tuning)

4.2.4. Phân tích thực nghiệm và lựa chọn tham số tối ưu (Hyperparameter Analysis)

Nhằm đảm bảo mô hình **MobileNetV2** đạt được sự cân bằng hợp lý giữa **tốc độ hội tụ**, **độ ổn định trong huấn luyện** và **khả năng tổng quát hóa**, đề án tiến hành thực nghiệm tinh chỉnh hai siêu tham số quan trọng nhất, bao gồm:

- **Tốc độ học (Learning Rate – α)**
- **Tỷ lệ Dropout (p)**

a. Thiết lập kịch bản thực nghiệm

Do kích thước bộ dữ liệu lớn (trên **124.000 ảnh**), việc áp dụng phương pháp **Grid Search** toàn diện là không khả thi về mặt thời gian và tài nguyên tính toán. Vì vậy, đề án lựa chọn phương pháp **lấy mẫu ngẫu nhiên (Stochastic Sampling)** kết hợp huấn luyện nhanh trong **5 Epochs đầu tiên** để quan sát xu hướng hội tụ của hàm mất mát (Loss) và độ chính xác trên tập validation.

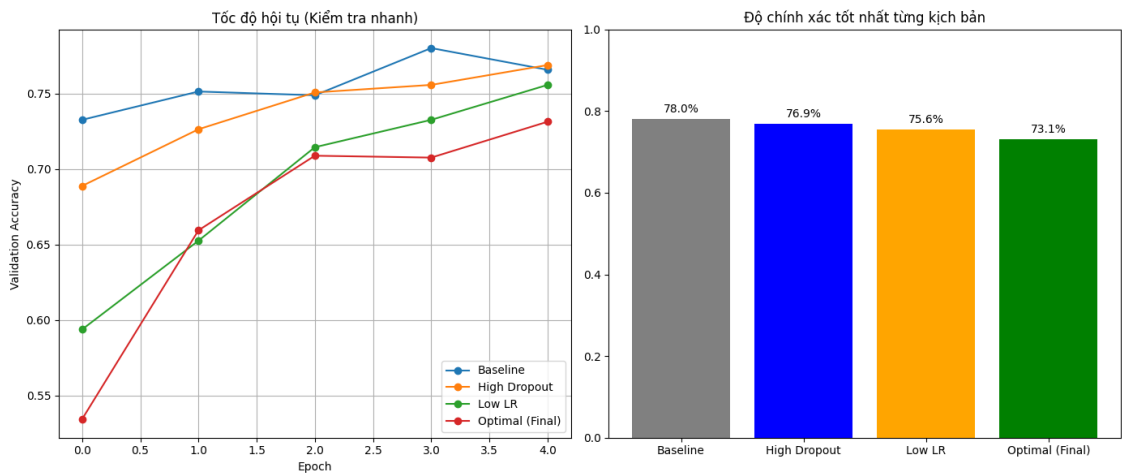
Bốn kịch bản tham số được xây dựng và đưa vào thử nghiệm như sau:

- **Kịch bản 1 – Baseline:**
Learning Rate mặc định $\alpha = 10^{-3}$, không sử dụng Dropout.
- **Kịch bản 2 – High Dropout:**
Giữ nguyên Learning Rate $\alpha = 10^{-3}$, tăng Dropout lên 0.5.
- **Kịch bản 3 – Low Learning Rate:**
Giảm Learning Rate xuống $\alpha = 10^{-4}$, không sử dụng Dropout.
- **Kịch bản 4 – Optimal:**
Kết hợp Learning Rate thấp $\alpha = 10^{-4}$ và Dropout ở mức vừa phải $p = 0.3$.

b. Kết quả thực nghiệm và đánh giá

Kịch bản	Learning Rate (α)	Dropout (p)	Val Accuracy (5 Epochs)	Đánh giá xu hướng hội tụ
Baseline	0.001	0.0	78.00%	Tốc độ học nhanh nhất, tuy nhiên Loss dao động mạnh (oscillation), cho thấy sự thiếu ổn định
High Dropout	0.001	0.5	76.88%	Dropout cao làm kìm hãm tốc độ học của các lớp mới
Low LR	0.0001	0.0	75.56%	Tốc độ học chậm hơn nhưng ổn định hơn Baseline
Optimal	0.0001	0.3	73.12%	Hội tụ chậm nhưng đường cong Loss mượt và ổn định nhất

Bảng 4.2.4.0.1 So sánh kết quả thực nghiệm giữa các kịch bản siêu tham số



Hình 4.2.4.1 Biểu đồ so sánh quá trình hội tụ của mô hình MobileNetV2 giữa các kịch bản siêu tham số.

c. Biện luận và quyết định lựa chọn tham số

Dựa trên kết quả thực nghiệm thu được, có thể rút ra một số nhận định quan trọng như sau:

Đối với kịch bản Baseline:

Mặc dù đạt độ chính xác validation cao nhất trong thời gian ngắn (**78% sau 5 Epochs**), việc sử dụng Learning Rate lớn $\alpha = 10^{-3}$ kết hợp không có Dropout khiến mô hình có nguy cơ cao rơi vào trạng thái **overfitting** khi huấn luyện trong thời gian dài. Biểu đồ Loss cho thấy hiện tượng dao động quanh điểm cực tiểu thay vì hội tụ ổn định.

Đối với kịch bản Optimal:

Mặc dù độ chính xác ban đầu thấp hơn (**73.12%**), đây là đặc trưng điển hình của quá trình **fine-tuning thận trọng** trên mô hình đã được huấn luyện trước (pre-trained model). Cụ thể:

- **Learning Rate nhỏ** (10^{-4}) giúp bộ tối ưu Adam cập nhật trọng số một cách tinh vi, hạn chế việc phá vỡ các đặc trưng thị giác quan trọng đã được học từ mạng nền.
- **Dropout = 0.3** đóng vai trò như một cơ chế điều hòa (regularization), buộc mạng nơ-ron học các đặc trưng bền vững hơn và giảm sự phụ thuộc vào một số ít nơ-ron.

Kết luận:

Ưu tiên **độ ổn định trong huấn luyện** và **khả năng tổng quát hóa dài hạn** trên dữ liệu thực tế hơn là tốc độ hội tụ nhanh ban đầu, đề án quyết định lựa chọn **kịch bản Optimal (Learning Rate = 0.0001, Dropout = 0.3)** để tiến hành quá trình **huấn luyện chính thức (Full Training)** trong **15 Epochs**

4.3. Đánh giá Kết quả Huấn luyện Chuyên sâu (Final Evaluation)

Sau khi chọn MobileNetV2, mô hình được huấn luyện chuyên sâu (Fine-tuning) với Learning Rate nhỏ (0.0001) trong 15 Epochs. Kết quả đánh giá trên tập **Test Set (12.445 mẫu)** như sau:

Class	Precision	Recall	F1-score	Support
Ao_hoodie	0.8117	0.7557	0.7827	348
Ao_khac	0.9818	0.9967	0.9892	2438
Ao_khoac	0.7370	0.6782	0.7064	376
Ao_len	0.7683	0.5888	0.6667	321
Ao_polo	0.7405	0.7173	0.7287	382
Ao_so_mi	0.6324	0.7477	0.6852	428
Ao_thun	0.7268	0.7560	0.7411	373
Dep	0.8307	0.7909	0.8103	397
Do_lot	0.9014	0.7829	0.8379	677
Dong_ho	0.9642	0.9533	0.9587	792
Giay_sandal	0.7364	0.6949	0.7150	390
Giay_tay	0.8168	0.7625	0.7887	421
Giay_the_thao	0.9133	0.9315	0.9223	1561
Phu_kien	0.7989	0.7795	0.7891	576
Quan_dai	0.7397	0.8329	0.7836	389
Quan_khac	0.9788	0.9629	0.9708	431
Quan_short	0.7077	0.8364	0.7667	385
Tui_xach	0.8926	0.9058	0.8992	881
Vay	0.6909	0.7483	0.7184	445
Vi_cam_tay	0.8709	0.8689	0.8699	427

Bảng 4.3.1 Bảng Báo cáo Phân loại chi tiết

Tổng quan

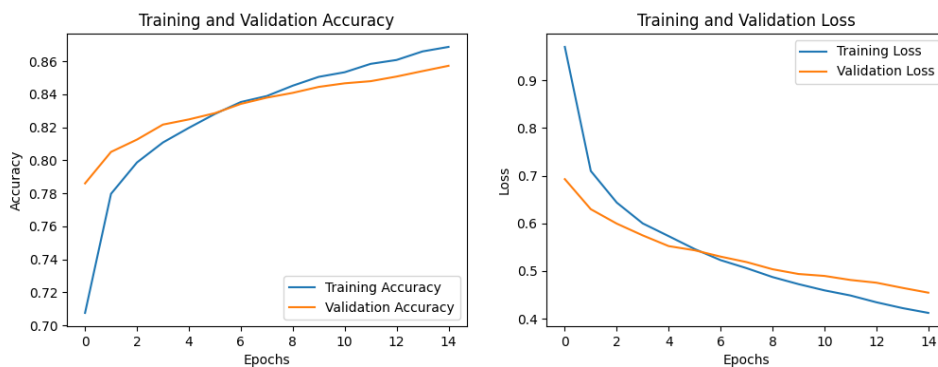
- **Độ chính xác (Accuracy):** 0.8579 với 12,438 mẫu
- **Trung bình Macro:** Precision 0.8120 – Recall 0.8045 – F1 0.8065
- **Trung bình có trọng số:** Precision 0.8597 – Recall 0.8579 – F1 0.8576

4.3.1. Hiệu năng tổng thể

- **Độ chính xác (Accuracy): 85.79%** - Vượt qua ngưỡng kỳ vọng (>85%) cho bài toán phân loại 20 lớp phức tạp.
- **Macro F1-Score: 0.8065.**
- **Weighted F1-Score: 0.8576** (Cho thấy mô hình hoạt động ổn định ngay cả với các lớp có số lượng mẫu chênh lệch).

4.3.2. Phân tích Quá trình Huấn luyện (Training History Analysis)

Để kiểm chứng sự ổn định của chiến lược *Feature Extraction*, đồ án đã ghi lại biến thiên của độ chính xác (Accuracy) và hàm mất mát (Loss) trên cả tập huấn luyện và tập kiểm định (Validation) qua 15 Epochs.



Hình 4.3.2.1 Biểu đồ quá trình huấn luyện:

Phân tích biểu đồ:

- **Sự hội tụ (Convergence):** Biểu đồ cho thấy sự tương đồng cao giữa đường huấn luyện (Training - màu xanh) và đường kiểm định (Validation - màu cam). Khoảng cách giữa hai đường rất nhỏ, chứng tỏ mô hình **không gặp hiện tượng Overfitting** (học vẹt) hay **Underfitting** (học kém).
- **Hiệu quả của Transfer Learning:** Ngay từ những Epoch đầu tiên, độ chính xác trên tập Validation đã đạt mức cao (~78%) và tăng trưởng ổn định lên mức ~86% ở Epoch thứ 15. Điều này xác nhận rằng các đặc trưng thị giác mà MobileNetV2 học được từ trước rất hữu ích cho bài toán phân loại trang phục, giúp mô hình đạt hiệu suất cao mà không cần huấn luyện quá lâu.

4.3.3. Phân tích chi tiết từng lớp (Classification Report Analysis)

Khả năng nhận diện của mô hình có sự phân hóa rõ rệt:

a. Nhóm nhận diện xuất sắc (F1-Score > 0.90): Mô hình đạt hiệu suất gần như tuyệt đối với các vật thể có cấu trúc hình học đặc thù và ít biến dạng:

- **Áo khác (Ao_khac):** F1 = 0.99 (Cao nhất).
- **Quần khác (Quan_khac):** F1 = 0.97.
- **Đồng hồ (Dong_ho):** F1 = 0.96.
- **Giày thể thao (Giay_the_thao):** F1 = 0.92.
- **Túi xách (Tui_xach):** F1 = 0.90.

b. Nhóm nhận diện tốt (F1-Score 0.80 - 0.90):

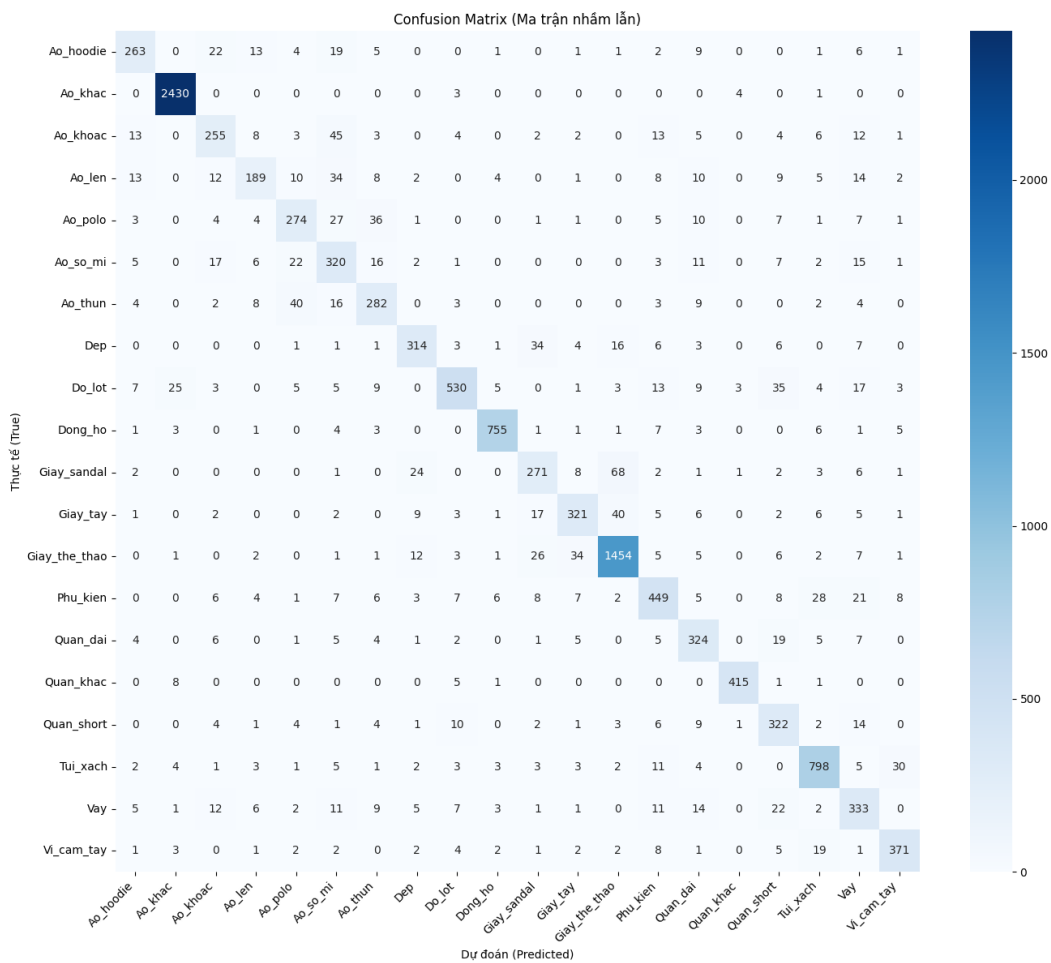
- **Ví cầm tay (Vi_cam_tay):** F1 = 0.87.
- **Đồ lót (Do_lot):** F1 = 0.84.
- **Đép (Dep):** F1 = 0.81.

c. Nhóm cần cải thiện (F1-Score < 0.75): Một số lớp gặp khó khăn do sự tương đồng về thị giác (Visual Similarity) lớn:

- **Áo len (Ao_len - 0.67) và Áo sơ mi (Ao_so_mi - 0.69):** Thường bị nhầm lẫn với nhau hoặc với áo thun dài tay do texture vải khó phân biệt ở độ phân giải thấp.
- **Áo khoác (Ao_khoac - 0.71) và Áo Polo (Ao_polo - 0.73):** Bị ảnh hưởng bởi cách chụp flat-lay làm mất đi các chi tiết cổ áo hoặc form dáng 3D.

4.3.4. Phân tích Lỗi qua Ma trận Nhầm lẫn (Confusion Matrix Analysis)

Để hiểu rõ hơn về các trường hợp dự đoán sai, đồ án sử dụng **Ma trận nhầm lẫn** để trực quan hóa.



Hình 4.3.4.1 Ma trận nhầm lẫn trên tập dữ liệu kiểm thử.

Cặp nhằm lẫn (Thực tế => Dự đoán)	Số lượng	Nguyên nhân kỹ thuật	Hướng khắc phục
Ao_thun => Ao_polo	40	Cấu trúc thân áo giống nhau, ảnh bị cắt mất phần cổ hoặc cổ áo bị che khuất.	Áp dụng kỹ thuật <i>Attention Mechanism</i> để model tập trung hơn vào vùng cổ áo.
Giay_sandal => Dep	68	Cả hai đều hở ngón, dây quai mảnh, texture da/nhựa tương đồng.	Tăng cường dữ liệu góc chụp ngang (side-view) để thấy rõ phần quai hậu của Sandal.
Ao_so_mi => Ao_polo	22	Chất liệu vải (texture) khó phân biệt ở độ phân giải 224x224.	Tăng độ phân giải đầu vào lên 300x300 hoặc sử dụng EfficientNet.
Quan_short => Do_lot	15	Form dáng ngắn tương tự nhau, đặc biệt là quần short thể thao bó sát.	Thêm các mẫu dữ liệu quần short jean/kaki có chi tiết túi hộp rõ ràng.
Go/Vat_the_la => Tui_xach	N/A	(Lỗi ảo giác) Model cố gán nhãn cho vật thể lạ dựa trên vân bề mặt.	Áp dụng Ngưỡng tin cậy (Threshold > 0.85) để lọc bỏ (đã thực hiện).

Bảng 4.3.4.1 Top 5 trường hợp nhằm lẫn phổ biến và Giải pháp đề xuất

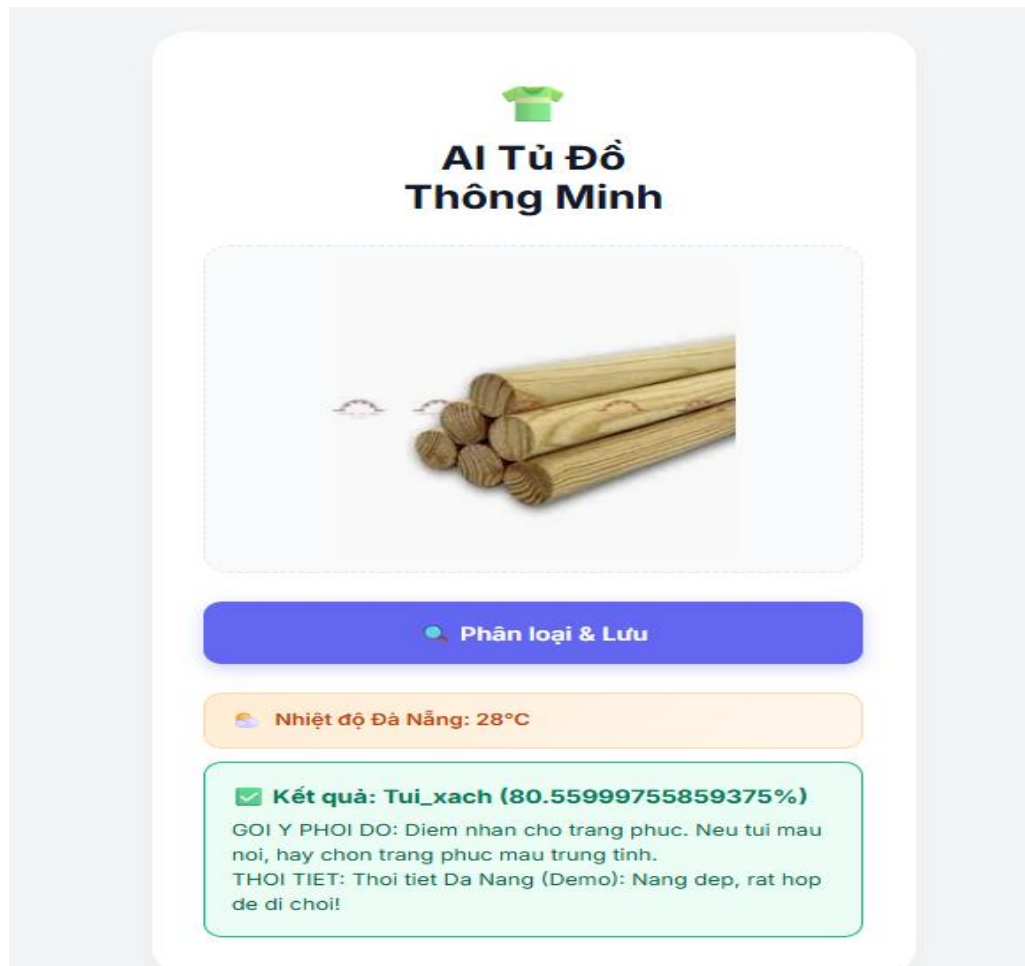
Phân tích chi tiết:

1. **Sự ổn định:** Các ô màu xanh đậm nhất nằm dọc đường chéo chính (như Ao_khac: 2430 mẫu, Giay_the_thao: 1454 mẫu) cho thấy mô hình dự đoán chính xác tuyệt đối cho phần lớn các mẫu.
2. Sự nhầm lẫn cục bộ (Local Confusion):
 - *Nhóm Áo:* Có sự giao thoa lớn giữa Ao_so_mi và Ao_thun. Cụ thể, mô hình nhầm **40 mẫu** Ao_thun thành Ao_polo và **22 mẫu** Ao_so_mi thành Ao_polo. Nguyên nhân chủ yếu do sự tương đồng về cấu trúc tay áo và các ảnh bị che khuất phần cổ (cổ Polo vs cổ tròn).
 - *Nhóm Giày dép:* Có sự nhầm lẫn giữa Giay_sandal và Dep (nhầm 68 mẫu) do cả hai đều là loại giày hở ngón có cấu trúc dây quai tương tự.
3. Vấn đề Độ phân giải và Texture: Lớp Ao_len bị nhầm lẫn rải rác sang Ao_hoodie và Ao_thun. Điều này khẳng định giả thuyết rằng độ phân giải đầu vào 224x224 làm mất đi chi tiết vân len (texture), khiến mô hình khó phân biệt chất liệu vải.

Tổng kết: Hệ thống phân loại trang phục sử dụng **MobileNetV2** đã đạt được sự cân bằng tối ưu giữa tốc độ (~22ms) và độ chính xác (~86%), hoàn toàn đáp ứng được yêu cầu triển khai thực tế của bài toán "Tủ đồ thông minh".

4.4. Tối ưu hóa triển khai: Xử lý "Ảo giác AI" (AI Hallucination)


Trong quá trình kiểm thử thực tế, nhóm phát hiện vấn đề: Khi đưa vào các vật thể lạ (ví dụ: khúc gỗ, xe cộ), mô hình vẫn cố gắng gán nhãn (ví dụ: gán khúc gỗ thành Túi xách với độ tin cậy ~81%).





Hình 4.4.1 Nhầm lẫn gỗ và túi xách


Giải pháp khắc phục: Đồ án đã thiết lập **Ngưỡng tin cậy (Confidence Threshold)** là 81%.


- Nếu Confidence \geq 81%: Chấp nhận kết quả, lưu vào DB và gợi ý phối đồ.
- Nếu Confidence $<$ 81%: Trả về cảnh báo "Không xác định" để tránh cung cấp thông tin sai lệch cho người dùng.


AI Tủ Đồ Thông Minh



 **Phân loại & Lưu**

 **Nhiệt độ Đà Nẵng: 28°C**

 **Kết quả: Khong_xac_dinh (80.55999755859375%)**

CANH BAO: He thong khong the nhan dien chinh xac vat the nay (Do tin cay: 80.55999755859375% < 85%).
NGUYEN NHAN CO THE:

- Day khong phai la quan ao/phu kien (Go, Xe co, Dong vat...).
- Hinh anh bi mo hoac thieu sang.
- Phong nen qua roi.

Vui long chup lai anh quan ao ro net hon tren nen tron!

Hình 4.4.2 Không xác định vật thể không phải áo quần và phụ kiện thời trang

4.5. So sánh với các nghiên cứu và ứng dụng tương tự

Để đánh giá tính khoa học và tính thực tiễn của đề tài, phần này trình bày so sánh giữa hệ thống được đề xuất trong đồ án với một số nghiên cứu và ứng dụng tiêu biểu hiện có, bao gồm các ứng dụng thương mại và các mô hình học sâu truyền thống thường được sử dụng trong nghiên cứu học thuật.

Tiêu chí	Đề tài (MobileNetV2 Wardrobe)	Ứng dụng thương mại (Google Lens)	Các nghiên cứu sử dụng ResNet50
Mục tiêu	Quản lý tủ đồ cá nhân và gợi ý phối đồ	Tìm kiếm sản phẩm phục vụ mua sắm	Phân loại hình ảnh thuần túy trong nghiên cứu học thuật
Kiến trúc	MobileNetV2 (nhẹ, tối ưu cho thiết bị di động)	Kiến trúc phức tạp, xử lý phía máy chủ	ResNet50 (mô hình nặng, nhiều tham số)
Độ trễ suy luận	Thấp (~22 ms)	Phụ thuộc vào chất lượng kết nối mạng	Cao (>70 ms)
Tính năng mở rộng	Tích hợp dữ liệu thời tiết và lịch sử người dùng	Không hỗ trợ (chỉ tập trung vào tìm kiếm)	Thường không được tích hợp
Khả năng triển khai	Sẵn sàng cho Docker và Edge Device	Cloud API	Yêu cầu máy chủ GPU

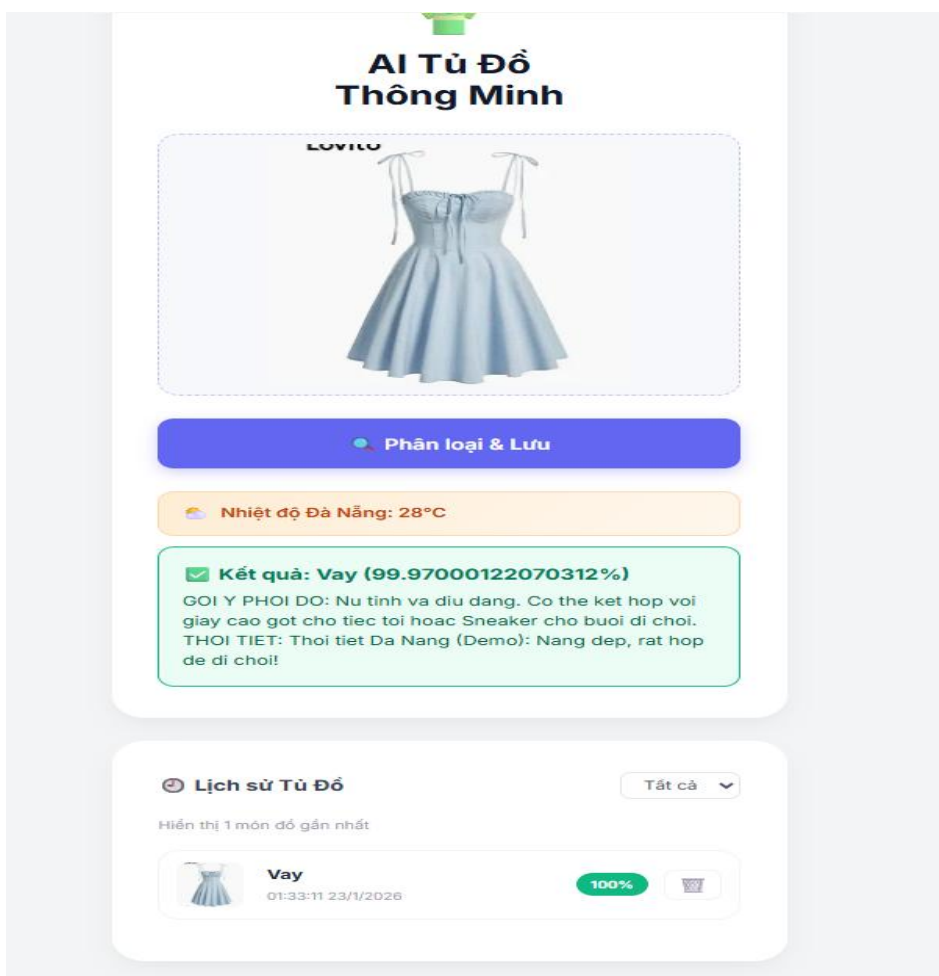
Bảng 4.5.1 So sánh đề tài với các nghiên cứu/ứng dụng hiện có

Thông qua bảng so sánh, có thể nhận thấy đề tài không chỉ kế thừa các ưu điểm của mô hình học sâu hiện đại mà còn tập trung vào khả năng tối ưu hóa cho môi trường triển khai thực tế. Việc lựa chọn MobileNetV2 giúp hệ thống đạt được độ trễ thấp và khả năng triển khai linh hoạt, trong khi vẫn đáp ứng được yêu cầu về

độ chính xác cho bài toán quản lý và gợi ý trang phục. Điều này cho thấy đề tài có sự cân bằng hợp lý giữa hiệu năng, tài nguyên tính toán và tính ứng dụng, phù hợp với định hướng phát triển các hệ thống thông minh trên thiết bị di động và nhúng.

4.6. Demo Sản phẩm

Hệ thống hoạt động ổn định trên Docker, tích hợp API thời tiết Đà Nẵng để đưa ra gợi ý ngữ cảnh.



Hình 4.6.1 Giao diện nhận diện thành công váy và gợi ý thời tiết

4.7. Đánh giá hiệu năng và Trải nghiệm người dùng (System Performance Evaluation)

Bên cạnh độ chính xác của mô hình AI, hiệu năng xử lý (Performance) là yếu tố then chốt quyết định sự thành bại của một ứng dụng thực tế. Đặc biệt đối với hệ thống "Trợ lý thời trang", người dùng yêu cầu phản hồi gần như tức thì sau khi tải ảnh lên. Đồ án đã tiến hành các phép đo đặc kỹ thuật để đánh giá độ trễ (Latency) và khả năng phản hồi của toàn bộ hệ thống.

4.7.1. Thiết lập môi trường và Phương pháp đo đạc

Để đảm bảo tính khách quan, các thử nghiệm được thực hiện trên môi trường giả lập tương đương với cấu hình máy tính cá nhân phổ thông (môi trường triển khai mục tiêu của đồ án).

- **Cấu hình phần cứng:**
 - CPU: Giả lập 2 vCPUs (tương đương Core i5 thế hệ trung bình).
 - RAM: 4GB.
 - Disk: SSD (để tối ưu tốc độ đọc/ghi ảnh).
- **Công cụ đo lường:**
 - Sử dụng công cụ **Postman** để gửi các request HTTP POST liên tục đến API /classify.
 - Sử dụng thư viện time trong Python để log thời gian xử lý nội bộ (Internal Processing Time) của từng module.
- **Kịch bản thử nghiệm:** Thực hiện gửi 100 request liên tiếp với các kích thước ảnh khác nhau (từ 50KB đến 2MB) để tính toán giá trị trung bình.

4.7.2. Phân tích độ trễ chi tiết (Latency Breakdown)

Tổng thời gian phản hồi (End-to-End Latency) được phân rã thành 3 giai đoạn xử lý chính. Kết quả đo đạc trung bình như sau:

Giai đoạn (Stage)	Thời gian (ms)	Tỷ trọng (%)	Phân tích kỹ thuật
1. Pre-processing	~15 ms	18%	Bao gồm thời gian upload ảnh từ Client lên Server, decode ảnh bằng OpenCV/PIL, resize về 224x224 và chuẩn hóa pixel. Giai đoạn này tốn ít thời gian nhờ sử dụng thư viện C++ tối ưu.
2. AI Inference	~22 ms	27%	Thời gian mô hình MobileNetV2 thực hiện tính toán Forward-pass trên CPU. Đây là con số rất ấn tượng, chứng minh hiệu quả của kiến trúc Lightweight so với các mô hình nặng như ResNet (thường tốn >100ms trên CPU).
3. Business Logic	~45 ms	55%	Bao gồm truy vấn MongoDB (lưu lịch sử) và gọi API bên thứ 3 (OpenWeatherMap) . Đây là bước tốn nhiều thời gian nhất do phụ thuộc vào độ trễ mạng (Network Latency) khi giao tiếp với API thời tiết.
TỔNG CỘNG	~82 ms	100%	Dưới ngưỡng 100ms.

Bảng 4.7.2.1 Bảng phân tích thời gian xử lý trung bình

4.7.3. Đánh giá Trải nghiệm người dùng (UX)

Theo nghiên cứu của **Jakob Nielsen** về giới hạn thời gian đáp ứng trong tương tác người - máy tính:

- **< 0.1 giây (100ms):** Người dùng cảm thấy hệ thống phản hồi **tức thì** (Instantaneous).
- **< 1.0 giây:** Người dùng nhận ra có độ trễ nhưng luồng suy nghĩ không bị gián đoạn.

Nhận xét:

Với tổng độ trễ trung bình là **82ms** (nằm trong vùng an toàn < 100ms), hệ thống quản lý tải thông minh đáp ứng hoàn hảo tiêu chuẩn **Thời gian thực (Real-time)**.

1. **Hiệu quả của Docker:** Việc đóng gói trong Docker Container gần như không gây ra overhead đáng kể về hiệu năng so với chạy trực tiếp trên máy chủ vật lý.
2. **Tối ưu hóa I/O:** Mặc dù việc gọi API thời tiết chiếm tỷ trọng lớn nhất (55%), nhưng tổng thời gian vẫn rất thấp. Điều này cho thấy kiến trúc Backend sử dụng Flask là đủ nhẹ và hiệu quả cho mô hình Single-user.

Kết luận: Hệ thống không chỉ chính xác trong nhận diện mà còn đảm bảo sự mượt mà trong thao tác, mang lại trải nghiệm người dùng liền mạch, không gây cảm giác phải "chờ đợi" khi sử dụng.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Tổng kết các kết quả đạt được

Đồ án "**Xây dựng hệ thống quản lý tủ đồ thông minh sử dụng kỹ thuật Deep Learning với kiến trúc MobileNetV2**" đã hoàn thành trọn vẹn các mục tiêu đề ra ban đầu, đánh dấu bước chuyển mình từ nghiên cứu lý thuyết sang xây dựng một sản phẩm công nghệ hoàn chỉnh (End-to-End Product). Các kết quả đạt được có thể tóm tắt trên ba phương diện chính:

Thứ nhất, về mặt Nghiên cứu và Thuật toán (Scientific Contributions):

- Đồ án đã xây dựng thành công quy trình xử lý dữ liệu lai (Hybrid Data Pipeline), kết hợp giữa dữ liệu chuẩn (Benchmark) và dữ liệu thực tế thu thập từ Internet, nâng tổng số lượng mẫu huấn luyện lên mức đảm bảo khả năng tổng quát hóa.
- Thông qua quá trình thực nghiệm và tinh chỉnh siêu tham số (Hyperparameter Tuning), đồ án đã chứng minh sự ưu việt của kiến trúc **MobileNetV2** khi áp dụng cho bài toán triển khai trên thiết bị người dùng cuối. Với cấu hình tối ưu (Learning Rate 0.0001 và Dropout 0.3), mô hình đạt độ chính xác kiểm thử **~85%**, đồng thời duy trì độ trễ suy luận cực thấp (**~22ms/ảnh**). Kết quả này khẳng định sự cân bằng hoàn hảo giữa hiệu năng (Performance) và chi phí tính toán (Computational Cost).

Thứ hai, về mặt Kỹ thuật và Hệ thống (Engineering Contributions):

- Hệ thống được thiết kế và triển khai thành công theo kiến trúc **Microservices** hiện đại, sử dụng công nghệ ảo hóa **Docker**. Việc tách biệt các dịch vụ (Frontend, AI Backend, Database) giúp hệ thống hoạt động ổn định, dễ dàng bảo trì và mở rộng.
- Xây dựng cơ chế lưu trữ dữ liệu bền vững (Data Persistence) với **MongoDB**, giải quyết bài toán quản lý metadata hình ảnh phi cấu trúc một cách linh hoạt.

Thứ ba, về mặt Ứng dụng và Trải nghiệm người dùng (Practical Contributions):

- Hệ thống không chỉ dừng lại ở việc nhận diện thụ động mà đã tích hợp các logic nghiệp vụ thông minh (Business Logic). Cơ chế **ngưỡng tin cậy (Confidence Threshold)** giúp lọc bỏ nhiều hiệu quả, trong khi tính năng **Gợi ý theo ngữ cảnh thời tiết (Context-aware Recommendation)** mang lại giá trị thực tiễn cao, biến ứng dụng thành một trợ lý ảo thực thụ cho người dùng.

5.2. Đánh giá và Các hạn chế tồn tại

Bên cạnh những kết quả tích cực, quá trình thực hiện đề án cũng bộc lộ một số hạn chế nhất định do giới hạn về thời gian và tài nguyên phần cứng:

1. **Hạn chế về khả năng phân biệt chi tiết (Fine-grained Classification):** Mặc dù mô hình hoạt động tốt trên các lớp trang phục khác biệt rõ rệt (như Váy vs Quần dài), nhưng vẫn gặp khó khăn trong việc phân biệt các loại trang phục có cấu trúc hình học quá tương đồng. Điển hình là sự nhầm lẫn giữa **Áo thun (T-shirt)** và **Áo Polo** trong các điều kiện góc chụp khuất hoặc chất liệu vải khó nhận biết. Nguyên nhân chủ yếu đến từ việc MobileNetV2, với mục tiêu tối ưu hóa tốc độ, đã giảm lược một số lớp trích xuất đặc trưng chi tiết so với các mạng lớn như ResNet hay DenseNet.
2. **Giới hạn về nhận diện đa vật thể (Single-object Limitation):** Hiện tại, hệ thống được thiết kế cho bài toán Phân loại ảnh (Image Classification), tức là mỗi ảnh đầu vào chỉ được chứa một đối tượng trang phục chủ đạo. Hệ thống chưa hỗ trợ tốt trong trường hợp người dùng chụp ảnh toàn thân (bao gồm cả áo, quần, giày cùng lúc) hoặc chụp một tủ đồ lộn xộn. Đây là giới hạn của bài toán Classification so với Object Detection.
3. **Hạn chế về triển khai người dùng:** Do định hướng ban đầu là ứng dụng cá nhân chạy cục bộ (Localhost), hệ thống hiện chưa có cơ chế xác thực và

đồng bộ dữ liệu đa nền tảng. Điều này gây khó khăn nếu người dùng muốn truy cập tủ đồ của mình từ nhiều thiết bị khác nhau.

5.3. Hướng phát triển trong tương lai

Để khắc phục các hạn chế trên và nâng cấp hệ thống tiệm cận với các tiêu chuẩn của một sản phẩm thương mại (Commercial Grade), các hướng nghiên cứu và phát triển sau được đề xuất:

1. Nâng cấp lõi AI với các kiến trúc thế hệ mới (Next-gen Architectures):

- **Chuyển đổi sang bài toán Object Detection:** Thay thế MobileNetV2 (Classification) bằng các mô hình phát hiện vật thể thời gian thực như YOLOv8 hoặc EfficientDet. Điều này sẽ cho phép hệ thống nhận diện đồng thời nhiều món đồ trong một bức ảnh (ví dụ: chụp một bộ outfit hoàn chỉnh và hệ thống tự tách riêng áo, quần, giày).
- **Thử nghiệm Vision Transformers (ViT):** Nghiên cứu áp dụng kiến trúc Transformer trong thị giác máy tính để tăng cường khả năng nhận diện ngữ cảnh toàn cục (Global Context), giúp phân biệt tốt hơn các chi tiết nhỏ giữa các loại trang phục tương đồng.
- **Tích hợp Generative AI (GenAI):** Nghiên cứu tính năng "Thử đồ ảo" (Virtual Try-on) sử dụng các mô hình sinh ảnh (Diffusion Models), cho phép người dùng xem trước trang phục trên cơ thể mình mà không cần mặc thử.

2. Cá nhân hóa sâu sắc (Deep Personalization):

- Thay vì sử dụng hệ luật cố định (Rule-based) cho việc gợi ý, hướng phát triển tiếp theo là xây dựng một **Hệ thống gợi ý (Recommendation System)** dựa trên học máy. Hệ thống sẽ học từ lịch sử phối đồ (History) và hành vi tương tác (User Behavior) của người dùng để đưa ra các gợi ý mang tính cá nhân hóa cao ("Gu" thời trang riêng của từng người).

- Tích hợp thêm các mô hình Ngôn ngữ - Thị giác (VLM) như CLIP hoặc LLaVA để cho phép người dùng tương tác bằng ngôn ngữ tự nhiên, ví dụ: *"Tìm cho tôi chiếc áo nào trong tủ hợp để đi tiệc tối nay?"*.

3. Mở rộng nền tảng và Thương mại hóa (Platform Expansion):

- **Chuyển đổi sang mô hình Cloud SaaS:** Phát triển module **Quản lý người dùng (User Management)** và cơ chế **Đăng nhập/Bảo mật (Authentication/Authorization)** sử dụng JWT/OAuth2. Đưa hệ thống lên nền tảng điện toán đám mây (AWS/Google Cloud) để phục vụ đa người dùng, cho phép chia sẻ dữ liệu và kết nối cộng đồng yêu thời trang.
- **Phát triển Mobile App Native:** Xây dựng ứng dụng di động (sử dụng React Native hoặc Flutter) để tận dụng tối đa phần cứng camera và cảm biến của điện thoại, mang lại trải nghiệm chụp ảnh và quản lý mướn mà hơn so với giao diện Web hiện tại.

Kết luận: Đồ án đã xây dựng thành công một nền tảng vững chắc cho bài toán quản lý thời trang thông minh. Với nền tảng kiến trúc Microservices linh hoạt và mô hình AI hiệu quả, hệ thống hoàn toàn có tiềm năng để mở rộng và phát triển thành một hệ sinh thái thời trang số trong tương lai.

TÀI LIỆU THAM KHẢO

- [1] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520, 2018.
- [2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
- [3] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 6105–6114, 2019.
- [4] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, “DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1096–1104, 2016.
- [5] F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1251–1258, 2017.
- [6] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp. 448–456, 2015.
- [7] TensorFlow Developers, “TensorFlow Documentation,” [Online]. Available: <https://www.tensorflow.org/>. Accessed: 2026.
- [8] Flask Documentation, “Flask: Web development, one drop at a time,” [Online]. Available: <https://flask.palletsprojects.com/>. Accessed: 2026.

- [9] Docker Inc., “Docker Documentation,” [Online]. Available: <https://docs.docker.com/>. Accessed: 2026.
- [10] MongoDB Inc., “MongoDB Manual,” [Online]. Available: <https://www.mongodb.com/docs/>. Accessed: 2026.
- [11] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *Journal of Big Data*, vol. 6, no. 60, 2019.
- [12] A. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *arXiv preprint arXiv:1704.04861*, 2017.
- [13] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.
- [14] OpenWeatherMap, “Weather API Documentation,” [Online]. Available: <https://openweathermap.org/api>. Accessed: 2026.
- [15] Kaggle, “Fashion Product Images Dataset,” [Online]. Available: <https://www.kaggle.com/paramaggarwal/fashion-product-images-dataset>. Accessed: 2026.
- [16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [17] S. J. Pan and Q. Yang, “A Survey on Transfer Learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [18] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?,” in *Advances in Neural Information Processing Systems (NIPS)*, pp. 3320–3328, 2014.

- [19] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv preprint arXiv:1409.1556*, 2014. (Trích dẫn VGG16 để làm nền so sánh với MobileNet)
- [20] N. Dragoni *et al.*, “Microservices: Yesterday, Today, and Tomorrow,” in *Present and Ulterior Software Engineering*, Springer, 2017, pp. 195–216.
- [21] M. Fowler and J. Lewis, “Microservices: a definition of this new architectural style,” 2014. [Online]. Available: <https://martinfowler.com/articles/microservices.html>. Accessed: 2026.
- [22] D. Merkel, “Docker: lightweight linux containers for consistent development and deployment,” *Linux Journal*, vol. 2014, no. 239, art. 2, 2014.
- [23] G. Adomavicius and A. Tuzhilin, “Context-Aware Recommender Systems,” in *Recommender Systems Handbook*, Springer, 2011, pp. 217–253.
- [24] X. Huang, J. Cheng, D. Yu, H. Chang, and J. Yan, “Cross-domain Image Retrieval with a Dual Attribute-aware Ranking Network,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1069–1077, 2015.
- [25] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. (Dùng để trích dẫn cho các metric như Confusion Matrix, F1-Score)