BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
NGÀNH KHOA HỌC MÁY TÍNH

Đề tài

NGHIÊN CỨU VÀ XÂY DỰNG API ĐỊNH DANH BỆNH NHÂN BẰNG NHẬN DIỆN KHUÔN MẶT VÀ ĐIỀU PHỐI HÀNG ĐỢI THỜI GIAN THỰC

RESEARCH AND DEVELOPMENT OF APIS FOR FACE RECOGNITION–BASED PATIENT IDENTIFICATION AND REAL-TIME QUEUE COORDINATION

Sinh viên: Nguyễn Hoàng Khánh
Mã số: B2113312
Khóa: K47
Giảng viên hướng dẫn: TS. Lưu Tiến Đạo

Cần Thơ, 12/2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
NGÀNH KHOA HỌC MÁY TÍNH

Đề tài

NGHIÊN CỨU VÀ XÂY DỰNG API ĐỊNH DANH BỆNH NHÂN BẰNG NHẬN DIỆN KHUÔN MẶT VÀ ĐIỀU PHỐI HÀNG ĐỢI THỜI GIAN THỰC

RESEARCH AND DEVELOPMENT OF APIS FOR FACE RECOGNITION–BASED PATIENT IDENTIFICATION AND REAL-TIME QUEUE COORDINATION

Giảng viên hướng dẫn
TS. Lưu Tiến Đạo

Sinh viên thực hiện
Nguyễn Hoàng Khánh
Mã số: B2113312
Khóa: K47

Cần Thơ, 12/2025

ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

XÁC NHẬN CHỈNH SỬA LUẬN VĂN THEO YÊU CẦU CỦA HỘI ĐỒNG

Tên luận văn (tiếng Việt và tiếng Anh): **Nghiên cứu và xây dựng API định danh bệnh nhân bằng nhận diện khuôn mặt và điều phối hàng đợi thời gian thực - Research and development of APIs for face recognition–based patient identification and real-time queue coordination**

Họ tên sinh viên: Nguyễn Hoàng Khánh MASV: B2113312

Mã lớp: DI21Z6A1

Đã báo cáo tại hội đồng ngành: Khoa học máy tính

Ngày báo cáo: 16/12/2025

Luận văn đã được chỉnh sửa theo góp ý của Hội đồng.

Cần Thơ, ngày ..... tháng …… năm 2025

Giáo viên hướng dẫn
(Ký và ghi họ tên)

TS. Lưu Tiến Đạo

NHẬN XÉT CỦA GIẢNG VIÊN

LỜI CẢM ƠN

Em xin gửi lời cảm ơn sâu sắc và chân thành nhất đến quý Thầy, Cô Trường Công nghệ Thông tin và Truyền thông – Trường Đại học Cần Thơ. Trong suốt thời gian học tập tại trường, em đã nhận được sự giảng dạy tận tâm, sự hỗ trợ quý báu và môi trường học tập chuyên nghiệp, giúp em tích lũy được những kiến thức và kỹ năng nền tảng cần thiết để thực hiện và hoàn thành luận văn tốt nghiệp này.

Em đặc biệt bày tỏ lòng biết ơn đến Giảng viên hướng dẫn – TS. Lưu Tiến Đạo, người đã tận tình đồng hành, định hướng và góp ý cho em trong suốt quá trình thực hiện đề tài. Sự hỗ trợ kiên nhẫn, tinh thần trách nhiệm và những nhận xét mang tính chuyên môn sâu sắc của Thầy đã giúp em vượt qua nhiều khó khăn, hoàn thiện phương pháp nghiên cứu và hoàn thành luận văn một cách tốt nhất.

Em cũng xin gửi lời cảm ơn chân thành đến gia đình và bạn bè, những người luôn là nguồn động viên lớn lao, khích lệ và tạo mọi điều kiện thuận lợi để em có thể yên tâm học tập và thực hiện đề tài.

Mặc dù đã nỗ lực hoàn thành luận văn với tinh thần trách nhiệm cao nhất, nhưng do hạn chế về thời gian và kinh nghiệm, chắc chắn bài luận văn vẫn không tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý của quý Thầy, Cô và mọi người để đề tài được hoàn thiện hơn trong tương lai.

Cuối cùng, em xin kính chúc quý Thầy, Cô luôn mạnh khỏe – hạnh phúc – thành công, và đạt nhiều thành tựu hơn nữa trong sự nghiệp giảng dạy và nghiên cứu.

Em xin chân thành cảm ơn!

Cần Thơ, ngày tháng 12 năm 2025

Người viết

Nguyễn Hoàng Khánh

MỤC LỤC

DANH MỤC TỪ VIẾT TẮT

STT	Viết tắt	Giải thích (Tiếng Việt)	Giải thích (Tiếng Anh)
1	AI	Trí tuệ nhân tạo	Artificial Intelligence
2	API	Giao diện lập trình ứng dụng	Application Programming Interface
3	CPU	Bộ xử lý trung tâm	Central Processing Unit
4	CUDA	Kiến trúc tính toán thống nhất cho GPU của NVIDIA	Compute Unified Device Architecture
5	EHR	Hồ sơ sức khỏe điện tử	Electronic Health Record
6	EMR	Hồ sơ y tế điện tử	Electronic Medical Record
7	ERD	Biểu đồ quan hệ thực thể	Entity–Relationship Diagram
8	ESP32-S3	Chip vi điều khiển ESP32-S3 dùng trong IoT/AIoT	ESP32-S3 Microcontroller
9	FAISS	Thư viện tìm kiếm vector hiệu năng cao của Facebook	Facebook AI Similarity Search
10	FCFS	Đến trước phục vụ trước	First-Come-First-Served
11	FIFO	Vào trước ra trước	First-In-First-Out
12	FK	Khóa ngoại	Foreign Key
13	GPU	Bộ xử lý đồ họa	Graphics Processing Unit
14	HIPAA	Đạo luật trách nhiệm & di động bảo hiểm y tế	Health Insurance Portability and Accountability Act
15	HIS	Hệ thống thông tin bệnh viện	Hospital Information System
16	HL7	Chuẩn trao đổi dữ liệu y tế cấp 7	Health Level Seven
17	IoT	Internet vạn vật	Internet of Things
18	MySQL	Hệ quản trị cơ sở dữ liệu quan hệ	MySQL Relational Database Management System
19	NVMe	Giao thức truy xuất bộ nhớ không bay hơi tốc độ cao	Non-Volatile Memory Express
20	ONNX	Chuẩn trao đổi mô hình mạng nơ-ron	Open Neural Network Exchange
21	PK	Khóa chính	Primary Key
22	RAM	Bộ nhớ truy cập ngẫu nhiên	Random Access Memory
23	SSD	Ổ đĩa trạng thái rắn	Solid State Drive
24	SQLite	Hệ quản trị CSDL nhẹ dạng nhúng	SQLite Embedded Database Engine
25	UML	Ngôn ngữ mô hình hóa thống nhất	Unified Modeling Language
26	Uvicorn	Máy chủ ASGI chạy ứng dụng FastAPI	Uvicorn ASGI Server
27	Vite	Công cụ build frontend	Vite Build Tool
28	Vue3	Khung giao diện Vue phiên bản 3	Vue.js 3 Framework
29	WebSocket	Giao thức kết nối thời gian thực	WebSocket Protocol

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

ABSTRACT

Traditional patient registration and queue management processes in hospitals rely heavily on manual operations, paper-based records, and human coordination, resulting in long waiting times, inefficient workflow, and a high risk of errors. In the context of accelerating digital transformation in healthcare, these limitations highlight the need for an automated, intelligent, and scalable system that improves the overall experience for both patients and medical staff

To address these challenges, this thesis presents the research, design, and implementation of a Face ID–Based Patient Record Management and Automated Queueing System, integrating advanced face recognition models with a real-time, two-layer queue coordination mechanism. The system utilizes the InsightFace framework, in which the ArcFace model is employed for 512-dimensional facial feature extraction and FAISS is used for high-performance vector retrieval. The backend is built on FastAPI, supporting asynchronous processing and WebSocket communication, while the frontend is developed with Vue3 to provide modern user interfaces for kiosk registration, operational dashboards, and administrative management. A key contribution of this thesis is the development of a dual-layer queue coordination algorithm, where the first layer categorizes patient requests into temporary queues based on selected services and specialties, and the second layer dynamically assigns patients to available doctors and rooms in real time, ensuring fairness (first-come–first-served), reducing congestion, and maintaining a consistent system-wide state.

Experimental evaluation demonstrates that the face recognition module achieves reliable identification performance in typical hospital lighting conditions, with an average processing time below 100 ms, while the queueing module successfully handles high-frequency requests and maintains real-time updates across multiple clients with a latency under 200 ms. The system has been fully implemented with core functionalities, including Face ID registration, multi-service queue management, doctor-side calling interfaces, and real-time dashboard monitoring. The results confirm that combining deep learning, optimized vector search, and modern software architecture provides a robust and practical solution for hospital digitalization, enhancing operational efficiency and improving patient experience, with strong potential for real-world deployment and future expansion.

Keywords: Face Recognition, ArcFace, InsightFace, FAISS, FastAPI, Queue Management, WebSocket, Hospital Digitalization

TÓM TẮT

Quy trình tiếp nhận và quản lý hàng chờ tại bệnh viện truyền thống phụ thuộc nhiều vào thao tác thủ công, giấy tờ và sự phối hợp của nhân viên, dẫn đến thời gian chờ đợi kéo dài, hiệu suất thấp và nguy cơ sai sót cao. Trong bối cảnh chuyển đổi số ngành y tế đang diễn ra mạnh mẽ, những hạn chế này đặt ra nhu cầu cấp thiết về một hệ thống tự động, thông minh và có khả năng mở rộng nhằm cải thiện trải nghiệm cho bệnh nhân cũng như hỗ trợ công tác vận hành của bệnh viện.

Nhằm giải quyết bài toán đó, luận văn trình bày quá trình nghiên cứu, thiết kế và xây dựng Hệ thống quản lý hồ sơ bệnh nhân bằng Face ID kết hợp điều phối hàng chờ khám bệnh tự động, tích hợp mô hình nhận diện khuôn mặt và thuật toán điều phối hai lớp theo thời gian thực. Hệ thống sử dụng thư viện InsightFace, trong đó mô hình ArcFace được áp dụng để trích xuất vector đặc trưng khuôn mặt 512 chiều và FAISS để tăng tốc tìm kiếm vector. Backend được xây dựng trên FastAPI hỗ trợ xử lý bất đồng bộ và giao tiếp WebSocket, trong khi frontend được phát triển bằng Vue3 để cung cấp các giao diện hiện đại gồm kiosk đăng ký, dashboard điều phối và trang quản trị. Đóng góp nổi bật của luận văn là đề xuất và hiện thực hóa thuật toán điều phối hai lớp, trong đó lớp thứ nhất phân loại yêu cầu khám vào hàng đợi tạm theo dịch vụ và chuyên khoa, còn lớp thứ hai tự động gán bệnh nhân vào phòng và bác sĩ phù hợp theo thời gian thực, đảm bảo tính công bằng và giảm tắc nghẽn.

Kết quả thực nghiệm cho thấy mô-đun Face ID đạt hiệu suất nhận diện ổn định trong điều kiện ánh sáng phòng khám, với thời gian xử lý trung bình dưới 100 ms, trong khi mô-đun điều phối hoạt động hiệu quả với tần suất yêu cầu cao và duy trì khả năng cập nhật thời gian thực đến các client với độ trễ dưới 200 ms. Hệ thống đã được xây dựng hoàn chỉnh với các chức năng chính như đăng ký bằng Face ID, quản lý hàng đợi đa dịch vụ, giao diện bác sĩ gọi bệnh nhân và dashboard giám sát theo thời gian thực. Các kết quả đạt được chứng minh rằng sự kết hợp giữa công nghệ học sâu, tìm kiếm vector tối ưu và kiến trúc phần mềm hiện đại có thể tạo ra một giải pháp số hóa bệnh viện mạnh mẽ và hiệu quả, đồng thời nâng cao hiệu suất vận hành và trải nghiệm bệnh nhân trong thực tế.

Từ khóa: Nhận diện khuôn mặt, ArcFace, InsightFace, FAISS, FastAPI, quản lý hàng chờ, WebSocket, chuyển đổi số y tế

I. PHẦN GIỚI THIỆU

1. Đặt vấn đề

Trong bối cảnh cuộc cách mạng công nghiệp 4.0 diễn ra mạnh mẽ, chuyển đổi số đã trở thành định hướng trọng tâm của nhiều quốc gia, trong đó có Việt Nam. Việc ứng dụng các công nghệ mới như trí tuệ nhân tạo (Artificial Intelligence – AI), Internet vạn vật (Internet of Things – IoT), thị giác máy tính (Computer Vision) và phân tích dữ liệu vào các lĩnh vực kinh tế – xã hội đang tạo ra những thay đổi sâu rộng trong cách thức tổ chức, quản lý và cung cấp dịch vụ. Các công nghệ này không chỉ góp phần nâng cao hiệu suất làm việc mà còn mở ra nhiều mô hình dịch vụ thông minh, linh hoạt và hiệu quả hơn.

Trong số các lĩnh vực chịu tác động mạnh mẽ của chuyển đổi số, y tế là một trong những lĩnh vực có tần suất tiếp xúc với người dân cao và yêu cầu độ chính xác, độ tin cậy rất lớn. Nhu cầu khám chữa bệnh ngày càng tăng, đặc biệt tại các bệnh viện tuyến tỉnh và tuyến trung ương, dẫn đến tình trạng quá tải thường xuyên trong khâu tiếp nhận và điều phối bệnh nhân. Ở nhiều cơ sở y tế, bệnh nhân phải xếp hàng chờ nhập thông tin, chờ xác nhận danh tính và chờ được phân vào phòng khám, gây mất thời gian và ảnh hưởng đến trải nghiệm khám chữa bệnh.

Mặc dù nhiều bệnh viện đã ứng dụng phần mềm hỗ trợ quản lý, quy trình tiếp nhận bệnh nhân trên thực tế vẫn mang tính bán thủ công. Nhân viên y tế phải thực hiện các thao tác kiểm tra thông tin, nhập dữ liệu, xác nhận hồ sơ và phân luồng bệnh nhân, dẫn đến nguy cơ sai sót và tắc nghẽn khi số lượng bệnh nhân tăng cao. Đặc biệt, trong các trường hợp bệnh nhân tái khám nhưng quên giấy tờ hoặc thông tin cá nhân không đầy đủ, quá trình tra cứu và xác nhận danh tính trở nên chậm chạp và kém hiệu quả.

Trong bối cảnh đó, công nghệ nhận diện khuôn mặt nổi lên như một giải pháp tiềm năng cho bài toán định danh bệnh nhân nhờ khả năng xác thực nhanh, chính xác và không yêu cầu mang theo giấy tờ. Khi kết hợp với hệ thống điều phối hàng đợi theo thời gian thực, quy trình tiếp nhận bệnh nhân có thể được tự động hóa ở mức cao hơn, giảm sự phụ thuộc vào thao tác thủ công và nâng cao hiệu quả vận hành. Tuy nhiên, để các giải pháp này có thể dễ dàng tích hợp vào hạ tầng công nghệ thông tin hiện có của bệnh viện, việc xây dựng các API đóng vai trò then chốt.

Xuất phát từ thực tiễn đó, đề tài "Nghiên cứu và xây dựng API cho hệ thống định danh bệnh nhân bằng nhận diện khuôn mặt và điều phối hàng đợi khám bệnh thời gian thực" được lựa chọn nhằm nghiên cứu, thiết kế và hiện thực hóa các API cốt lõi phục vụ bài toán tiếp nhận và điều phối bệnh nhân, làm nền tảng cho việc phát triển các hệ thống y tế thông minh trong tương lai.

2. Thực trạng quản lý hồ sơ bệnh nhân và bốc số khám bệnh hiện nay

Hiện nay, phần lớn các bệnh viện và phòng khám tại Việt Nam vẫn áp dụng quy trình tiếp nhận bệnh nhân theo hướng bán thủ công. Bệnh nhân cần cung cấp giấy tờ tùy thân và thông tin cá nhân để nhân viên y tế nhập vào hệ thống. Quy trình này tồn tại nhiều hạn chế như:

- Thời gian tiếp nhận kéo dài: mỗi bệnh nhân phải trải qua nhiều bước như khai báo thông tin, kiểm tra dữ liệu, xác nhận và xếp hàng chờ khám, dễ gây ùn tắc khi lượng bệnh nhân tăng cao.
- Sai sót trong nhập liệu: dữ liệu phụ thuộc nhiều vào thao tác thủ công của nhân viên, dễ xảy ra nhầm lẫn hoặc thiếu thông tin.
- Khó khăn trong việc định danh bệnh nhân tái khám: bệnh nhân quên giấy tờ hoặc thông tin không đầy đủ khiến việc tra cứu hồ sơ mất nhiều thời gian.
- Hệ thống xếp hàng chưa tối ưu: nhiều cơ sở sử dụng máy bốc số đơn lẻ, không liên kết chặt chẽ với dữ liệu bệnh nhân và không hỗ trợ điều phối linh hoạt.
- Thiếu cơ chế điều phối thời gian thực: khi một phòng khám quá tải hoặc có thay đổi đột xuất, việc điều chỉnh vẫn phải thực hiện thủ công, làm giảm hiệu quả vận hành.

Những hạn chế trên cho thấy nhu cầu cần có một giải pháp kỹ thuật tập trung vào việc tự động hóa định danh bệnh nhân và điều phối hàng đợi, đặc biệt ở mức API, nhằm tạo nền tảng cho các hệ thống y tế thông minh.

3. Vai trò của chuyển đổi số trong bệnh viện

Chuyển đổi số trong y tế không chỉ là xu hướng mà còn là yêu cầu tất yếu, được thể hiện trong các chiến lược chuyển đổi số quốc gia và định hướng của Bộ Y tế. Trong đó, các hệ thống API đóng vai trò quan trọng như một lớp trung gian kết nối giữa các thành phần khác nhau của hệ thống thông tin bệnh viện.

Cụ thể, hệ thống API mang lại các lợi ích sau:

- Chuẩn hóa và tự động hóa quy trình tiếp nhận: API cho phép các hệ thống khác nhau (kiosk, website, dashboard) truy cập và xử lý dữ liệu một cách thống nhất.
- Tăng tốc độ định danh bệnh nhân: tích hợp API Face ID giúp xác thực danh tính nhanh chóng, đặc biệt với bệnh nhân tái khám.
- Hỗ trợ điều phối hàng đợi linh hoạt: API thời gian thực cho phép cập nhật trạng thái khám và phân luồng bệnh nhân hiệu quả.
- Khả năng mở rộng và tích hợp: API có thể dễ dàng tích hợp với các hệ thống HIS, EHR hoặc các nền tảng y tế số khác trong tương lai.

4. Mục tiêu của đề tài

4.1. Mục tiêu tổng quát

Nghiên cứu và xây dựng một hệ thống API ứng dụng trí tuệ nhân tạo trong định danh bệnh nhân bằng nhận diện khuôn mặt, kết hợp cơ chế điều phối hàng đợi khám bệnh theo thời gian thực, nhằm hỗ trợ tự động hóa quy trình tiếp nhận bệnh nhân và nâng cao hiệu quả vận hành tại các cơ sở y tế.

4.2. Mục tiêu cụ thể

- Xây dựng API định danh bệnh nhân dựa trên công nghệ nhận diện khuôn mặt sử dụng các mô hình học sâu như ArcFace và MobileFaceNet.
- Thiết kế cơ sở dữ liệu phục vụ lưu trữ thông tin bệnh nhân, embedding Face ID và lịch sử khám ở mức phục vụ API.
- Phát triển API bốc số và điều phối hàng đợi khám bệnh theo thời gian thực dựa trên trạng thái phòng khám và bác sĩ.
- Xây dựng các thành phần prototype (kiosk và dashboard) nhằm kiểm thử và đánh giá khả năng hoạt động của API.
- Đánh giá hiệu năng và độ chính xác của hệ thống thông qua các kịch bản thử nghiệm mô phỏng thực tế..

5. Đối tượng và phạm vi nghiên cứu

5.1. Đối tượng nghiên cứu

Đối tượng nghiên cứu của luận văn là các API và thuật toán cốt lõi phục vụ định danh bệnh nhân bằng nhận diện khuôn mặt và điều phối hàng đợi khám bệnh theo thời gian thực, bao gồm mô hình Face ID, cơ chế hàng đợi và giao tiếp thời gian thực giữa các thành phần hệ thống.

5.2. Phạm vi nghiên cứu

Luận văn tập trung vào việc thiết kế và xây dựng API cho bài toán tiếp nhận và điều phối bệnh nhân. Đề tài không đi sâu vào các hệ thống bệnh án điện tử (EHR/EMR) hoặc các nghiệp vụ chuyên sâu khác của bệnh viện. Hệ thống được xây dựng ở mức prototype nhằm phục vụ mục đích nghiên cứu, kiểm thử và đánh giá khả năng ứng dụng.

6. Phương pháp nghiên cứu

Trong quá trình thực hiện luận văn, đề tài sử dụng kết hợp nhiều phương pháp nghiên cứu khác nhau nhằm đảm bảo tính khoa học, tính thực tiễn và khả năng triển khai của hệ thống đề xuất.

6.1. Phương pháp nghiên cứu tài liệu

Phương pháp nghiên cứu tài liệu được sử dụng nhằm xây dựng nền tảng lý thuyết cho đề tài. Các tài liệu tham khảo bao gồm các công trình nghiên cứu, bài báo khoa học và tài liệu kỹ thuật liên quan đến nhận diện khuôn mặt, hệ thống xếp hàng, trí tuệ nhân tạo và Internet of Things (AI/IoT), cũng như các mô hình và thư viện tiêu biểu như InsightFace và ArcFace. Thông qua việc tổng hợp và phân tích các tài liệu này, luận văn hình thành được cơ sở lý thuyết vững chắc để làm nền tảng cho quá trình thiết kế và triển khai hệ thống..

6.2. Phương pháp khảo sát thực tế

Phương pháp khảo sát thực tế được áp dụng nhằm đảm bảo đề tài bám sát nhu cầu và điều kiện triển khai trong môi trường bệnh viện. Quá trình khảo sát tập trung vào việc quan sát quy trình tiếp nhận bệnh nhân tại cơ sở y tế, từ đó xác định các điểm còn hạn chế trong phương thức vận hành hiện tại, các công đoạn có thể tự động hóa và các yêu cầu thực tế của nhân viên y tế. Kết quả khảo sát giúp định hướng thiết kế hệ thống theo hướng phù hợp với thực tiễn, tránh việc xây dựng giải pháp mang tính lý thuyết thuần túy.

6.3. Phương pháp phân tích – thiết kế hệ thống

Dựa trên các yêu cầu nghiệp vụ đã thu thập được, luận văn áp dụng phương pháp phân tích và thiết kế hệ thống để mô hình hóa bài toán một cách khoa học. Các mô hình UML, ERD, Use Case và Sequence Diagram được sử dụng nhằm mô tả cấu trúc dữ liệu, các thành phần chức năng và luồng xử lý của hệ thống. Cách tiếp cận này giúp hệ thống được thiết kế rõ ràng, dễ triển khai, đồng thời thuận lợi cho việc mở rộng và bảo trì trong tương lai..

6.4. Phương pháp lập trình và triển khai thử nghiệm

Phương pháp lập trình được sử dụng để hiện thực hóa các mô hình thiết kế thành một hệ thống hoạt động hoàn chỉnh. Backend của hệ thống được xây dựng bằng Python với framework FastAPI, frontend sử dụng Vue3, trong khi cơ sở dữ liệu được triển khai trên MySQL hoặc SQLite tùy theo môi trường thử nghiệm. Bên cạnh đó, các kỹ thuật AI/IoT được tích hợp nhằm xây dựng mô-đun nhận diện khuôn mặt và các thành phần giao tiếp với kiosk, dashboard và thiết bị ngoại vi. Việc triển khai thử nghiệm cho phép đánh giá trực tiếp khả năng vận hành của hệ thống trong các kịch bản mô phỏng thực tế.

6.5. Phương pháp kiểm thử và đánh giá

Phương pháp kiểm thử và đánh giá được sử dụng để xác minh hiệu quả và độ tin cậy của hệ thống. Quá trình kiểm thử được thực hiện ở cả mức đơn lẻ từng mô-đun và mức tổng thể toàn hệ thống. Các tiêu chí đánh giá chính bao gồm độ chính xác của mô-đun Face ID, thời gian xử lý của các chức năng bốc số và điều phối, tốc độ cập nhật dữ liệu theo thời gian thực và mức độ ổn định của hệ thống khi tải tăng cao. Đây

là bước quan trọng nhằm chứng minh tính khả thi và hiệu quả của giải pháp được đề xuất trong luận văn.

7. Ý nghĩa khoa học và ý nghĩa thực tiễn

7.1. Ý nghĩa khoa học

Luận văn đề xuất mô hình xây dựng API kết hợp giữa nhận diện khuôn mặt và điều phối hàng đợi theo thời gian thực, góp phần bổ sung cơ sở khoa học cho các nghiên cứu về hệ thống tiếp nhận bệnh nhân thông minh.

7.2. Ý nghĩa thực tiễn

Kết quả nghiên cứu có thể được sử dụng như nền tảng API để phát triển và tích hợp vào các hệ thống y tế trong thực tế, góp phần nâng cao hiệu quả tiếp nhận và điều phối bệnh nhân.

Phần giới thiệu

Giới thiệu tổng quát về đề tài.

Phần nội dung

Chương 1:Tổng quan về chuyển đổi số trong y tế và các giải pháp liên quan.

Chương 2: Phân tích bài toán và thiết kế hệ thống.

Chương 3: Xây dựng và triển khai hệ thống.

Chương 4: Thử nghiệm và đánh giá hệ thống.

Phần kết luận

Trình bày kết quả đạt được và hướng phát triển hệ thống.

Tài liệu tham khảo

II. PHẦN NỘI DUNG

CHƯƠNG 1: TỔNG QUAN VỀ CHUYỂN ĐỔI SỐ TRONG Y TẾ VÀ CÁC GIẢI PHÁP LIÊN QUAN

1.1. Chuyển đổi số trong lĩnh vực y tế

Chuyển đổi số trong y tế được hiểu là quá trình ứng dụng các công nghệ số như dữ liệu lớn (Big Data), trí tuệ nhân tạo (Artificial Intelligence – AI), điện toán đám mây (Cloud Computing), Internet vạn vật (Internet of Things – IoT) và các nền tảng kết nối vào hoạt động quản lý, vận hành và cung cấp dịch vụ y tế. Quá trình này không chỉ dừng lại ở việc số hóa hồ sơ giấy, mà hướng đến tái cấu trúc quy trình nghiệp vụ, chuẩn hóa dữ liệu và xây dựng các nền tảng kết nối linh hoạt nhằm nâng cao chất lượng phục vụ và hiệu quả vận hành của hệ thống y tế.

Trong những năm gần đây, Bộ Y tế đã ban hành nhiều chủ trương quan trọng liên quan đến chuyển đổi số, bám sát Chương trình Chuyển đổi số quốc gia (Quyết định 749/QĐ-TTg). Các chiến lược này nhấn mạnh mục tiêu xây dựng cơ sở dữ liệu y tế, triển khai hồ sơ sức khỏe điện tử, bệnh án điện tử và từng bước hình thành mô hình bệnh viện thông minh. Tuy nhiên, thực tế triển khai cho thấy tiến trình chuyển đổi số vẫn gặp nhiều khó khăn, đặc biệt tại các cơ sở y tế tuyến cơ sở, do hạn chế về hạ tầng công nghệ thông tin, nguồn nhân lực và khả năng tích hợp giữa các hệ thống.

Một trong những thách thức lớn của chuyển đổi số y tế hiện nay là sự thiếu đồng bộ giữa các phần mềm và hệ thống thông tin. Nhiều bệnh viện sử dụng các giải pháp rời rạc cho từng nghiệp vụ, gây khó khăn trong chia sẻ dữ liệu và làm giảm hiệu quả vận hành. Do đó, việc xây dựng các API đóng vai trò trung gian kết nối giữa các mô-đun chức năng và hệ thống khác nhau được xem là hướng tiếp cận phù hợp, giúp tăng tính linh hoạt, khả năng mở rộng và khả năng tích hợp của hệ thống y tế số.

Nhìn chung, chuyển đổi số trong y tế vừa mang tính kỹ thuật (hạ tầng CNTT, dữ liệu, phần mềm), vừa mang tính quản trị (tái thiết kế quy trình nghiệp vụ), trong đó các nền tảng API là thành phần quan trọng giúp hiện thực hóa các mô hình y tế thông minh theo hướng mở và bền vững.

1.1.1. Khái niệm bệnh viện thông minh, HIS, EHR

Bệnh viện thông minh (Smart Hospital) là mô hình bệnh viện trong đó các hoạt động chuyên môn và quản trị được hỗ trợ bởi các hệ thống thông tin số, có khả năng kết nối, chia sẻ dữ liệu và tự động hóa một phần quy trình vận hành. Thay vì sử dụng các phần mềm độc lập cho từng nghiệp vụ, bệnh viện thông minh hướng tới xây dựng một hệ sinh thái số, trong đó các hệ thống được liên thông thông qua các chuẩn dữ liệu và giao diện lập trình ứng dụng (API).

Trong hệ sinh thái đó, HIS (Hospital Information System) là hệ thống quản lý thông tin bệnh viện, đảm nhiệm các nghiệp vụ hành chính và vận hành như tiếp nhận bệnh nhân, quản lý thông tin hành chính, chỉ định dịch vụ, viện phí và báo cáo thống kê. HIS thường được xem là hệ thống trung tâm, cung cấp dữ liệu cho các mô-đun và hệ thống liên quan thông qua các cơ chế tích hợp.

EHR (Electronic Health Record) là hồ sơ sức khỏe điện tử, lưu trữ thông tin y tế của người bệnh trong suốt quá trình chăm sóc sức khỏe. EHR hướng đến khả năng liên thông dữ liệu giữa nhiều cơ sở y tế, hỗ trợ theo dõi liên tục và toàn diện tình trạng sức khỏe của bệnh nhân. Tuy nhiên, việc triển khai EHR ở quy mô lớn đòi hỏi hạ tầng kỹ thuật, chuẩn dữ liệu và cơ chế bảo mật phức tạp.

Trong phạm vi nghiên cứu của luận văn này, các khái niệm HIS và EHR được xem xét ở góc độ hệ thống tích hợp, làm cơ sở tham chiếu để thiết kế các API phục vụ định danh bệnh nhân và điều phối hàng đợi, thay vì xây dựng đầy đủ các chức năng của HIS hoặc EHR.

1.1.2. Các mô-đun thường gặp: tiếp nhận – hồ sơ – xếp hàng – thanh toán

Trong các hệ thống thông tin y tế hiện đại, quy trình tiếp nhận và khám bệnh thường được tổ chức thành các mô-đun chức năng có mối liên hệ chặt chẽ với nhau.

Mô-đun tiếp nhận và định danh bệnh nhân (Registration & Identification) có nhiệm vụ ghi nhận thông tin ban đầu và xác thực danh tính bệnh nhân. Bên cạnh các phương thức truyền thống như nhập liệu thủ công hoặc quét mã QR, nhiều cơ sở y tế đã bắt đầu nghiên cứu và ứng dụng các phương thức định danh sinh trắc học, trong đó nhận diện khuôn mặt là một giải pháp tiềm năng nhờ tốc độ xác thực nhanh và giảm sự phụ thuộc vào giấy tờ.

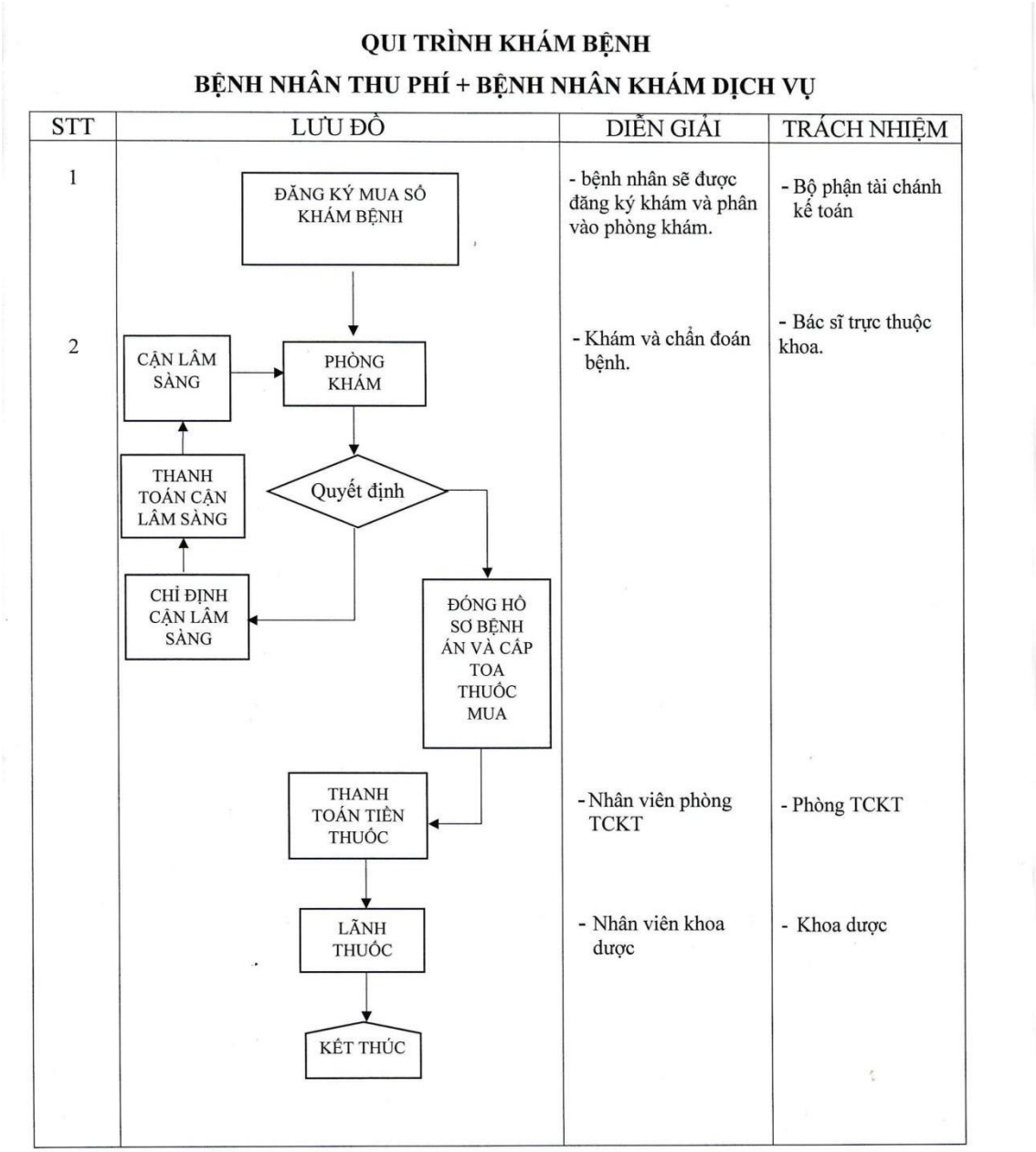
Mô-đun xếp hàng và điều phối bệnh nhân (Queue Management) chịu trách nhiệm tổ chức bốc số, gọi lượt khám và phân luồng bệnh nhân giữa các phòng khám. Các hệ thống xếp hàng hiện đại không chỉ dừng lại ở việc cấp số thứ tự, mà còn hỗ trợ cập nhật trạng thái theo thời gian thực, cho phép nhân viên y tế theo dõi tình hình khám bệnh và chủ động điều phối khi xảy ra quá tải.

Mô-đun thanh toán và viện phí hỗ trợ tính toán chi phí dịch vụ và tích hợp các hình thức thanh toán điện tử. Trong nhiều hệ thống, mô-đun này được triển khai độc lập và kết nối với các mô-đun khác thông qua API.

Từ góc độ kỹ thuật, các mô-đun trên thường không hoạt động độc lập mà được liên kết với nhau thông qua các API, đóng vai trò là lớp trung gian trao đổi dữ liệu giữa kiosk tiếp nhận, hệ thống xếp hàng, dashboard quản lý và các hệ thống thông tin bệnh viện hiện có. Đây cũng chính là trọng tâm nghiên cứu của luận văn.

1.2. Hệ thống quản lý hồ sơ bệnh nhân và bốc số khám bệnh truyền thống

1.2.1. Quy trình tiếp nhận bệnh nhân thông thường



Hình 1.1 Minh họa quá trình tiếp nhận bệnh nhân thông thường

Hình 1.1 minh họa quy trình tiếp nhận và khám bệnh ngoại trú tại các cơ sở y tế chưa được số hóa toàn diện, nơi các hoạt động vẫn chủ yếu dựa trên giấy tờ và thao tác thủ công. Một quy trình tiếp nhận bệnh nhân điển hình bao gồm các bước sau:

- Tiếp nhận lượt khám ban đầu: Bệnh nhân đến khu vực tiếp nhận và được phát số thứ tự khám, thường thông qua sổ giấy hoặc máy lấy số đơn giản, chưa có sự liên thông dữ liệu với các hệ thống khác.

- Thu thập thông tin hành chính: Bệnh nhân xuất trình giấy tờ tùy thân, thẻ bảo hiểm y tế và cung cấp thông tin cá nhân; nhân viên tiếp nhận nhập lại dữ liệu thủ công vào sổ hoặc phần mềm nội bộ.
- Lập hoặc truy xuất thông tin bệnh nhân: Đối với bệnh nhân khám lần đầu, thông tin được ghi nhận mới; với bệnh nhân tái khám, nhân viên phải tra cứu hồ sơ cũ dựa trên các thông tin định danh truyền thống.
- Hướng dẫn đến phòng khám: Dựa trên chuyên khoa và dịch vụ đăng ký, bệnh nhân được chỉ định phòng khám phù hợp và nhận phiếu khám.
- Tạm ứng hoặc thanh toán chi phí (nếu có): Một số cơ sở yêu cầu bệnh nhân thực hiện các thủ tục tài chính trước khi vào khám hoặc làm cận lâm sàng.

Quy trình này khiến bệnh nhân phải di chuyển qua nhiều quầy chức năng, xếp hàng lặp lại và phụ thuộc nhiều vào thao tác thủ công của nhân viên y tế, dẫn đến thời gian chờ kéo dài và tiềm ẩn nguy cơ sai sót trong việc định danh và quản lý thông tin bệnh nhân.

1.2.2. Các hạn chế: thời gian chờ, sai sót, phụ thuộc giấy tờ, thiếu tự động hóa

Mô hình tiếp nhận và xếp hàng khám bệnh truyền thống bộc lộ nhiều hạn chế, bao gồm:

- Thời gian chờ đợi kéo dài: Bệnh nhân phải xếp hàng ở nhiều khâu khác nhau. Khi một khâu quá tải, toàn bộ quy trình dễ bị đình trệ.
- Dễ xảy ra sai sót trong nhập liệu và định danh: Việc phụ thuộc vào thao tác thủ công khiến thông tin cá nhân dễ bị nhập sai, dẫn đến nhầm lẫn hồ sơ hoặc chậm trễ trong quá trình khám.
- Phụ thuộc vào giấy tờ truyền thống: Hồ sơ giấy khó lưu trữ, khó tra cứu lịch sử dài hạn và gây tốn kém về không gian cũng như nhân lực.
- Thiếu liên kết và tự động hóa giữa các khâu: Máy bốc số, hệ thống tiếp nhận và các phần mềm nội bộ thường hoạt động rời rạc, dữ liệu không được đồng bộ theo thời gian thực.

Những hạn chế trên cho thấy nhu cầu cần có một giải pháp công nghệ tập trung vào tự động hóa định danh và điều phối hàng đợi, đóng vai trò làm lớp tích hợp giữa các thành phần trong quy trình tiếp nhận bệnh nhân.

1.3. Ứng dụng AI/IoT và Face ID trong bệnh viện

1.3.1. Khái quát về nhận diện khuôn mặt trong y tế

Nhận diện khuôn mặt là một bài toán thuộc lĩnh vực thị giác máy tính, trong đó ảnh khuôn mặt được biểu diễn dưới dạng vector đặc trưng (embedding) trong không

gian nhiều chiều. Các ảnh của cùng một người sẽ có embedding gần nhau, trong khi các ảnh của những người khác nhau sẽ có khoảng cách lớn hơn.

Trong lĩnh vực y tế, Face ID có thể được ứng dụng vào các khâu như:

- Định danh và xác nhận danh tính bệnh nhân trong quá trình tiếp nhận, hỗ trợ truy xuất thông tin thông qua các API tích hợp.
- Kiểm soát ra vào các khu vực nhạy cảm, như phòng mổ hoặc khoa hồi sức, nhằm nâng cao mức độ an ninh.
- Hỗ trợ theo dõi bệnh nhân nội trú, hạn chế nhầm lẫn danh tính trong quá trình chăm sóc.

Các nghiên cứu gần đây cho thấy, với các mô hình học sâu hiện đại, hệ thống nhận diện khuôn mặt có thể đạt độ chính xác cao và phù hợp để sử dụng như một phương thức định danh hỗ trợ trong môi trường y tế.

1.3.2. Lợi ích của Face ID trong quản lý bệnh nhân

Khi được tích hợp vào hệ thống thông tin y tế thông qua các API, công nghệ Face ID mang lại nhiều lợi ích thiết thực. Trước hết, Face ID giúp rút ngắn thời gian tiếp nhận bệnh nhân, bởi quá trình xác thực danh tính có thể được thực hiện nhanh chóng mà không cần nhập liệu thủ công nhiều thông tin.

Bên cạnh đó, việc sử dụng đặc trưng khuôn mặt làm yếu tố định danh giúp giảm nguy cơ nhầm lẫn hồ sơ, đặc biệt trong trường hợp bệnh nhân có thông tin cá nhân trùng lặp. Face ID cũng góp phần nâng cao mức độ tự động hóa và khả năng liên thông dữ liệu, khi thông tin định danh có thể được truyền trực tiếp đến các mô-đun xếp hàng và điều phối khám bệnh.

Ngoài ra, Face ID phù hợp với xu hướng tiếp nhận không tiếp xúc (contactless), giúp giảm nguy cơ lây nhiễm chéo và nâng cao trải nghiệm của người bệnh trong môi trường bệnh viện.

1.4. Tổng quan một số mô hình nhận diện khuôn mặt

1.4.1. FaceNet

FaceNet (Google, 2015) là mô hình tiêu biểu trong hướng tiếp cận học embedding khuôn mặt. Thay vì phân loại trực tiếp từng danh tính, FaceNet ánh xạ ảnh khuôn mặt vào không gian vector (thường 128 chiều) sao cho các ảnh của cùng một người có khoảng cách gần nhau, trong khi ảnh của những người khác nhau có khoảng cách xa hơn.

Mô hình sử dụng triplet loss với ba ảnh đầu vào (anchor, positive, negative), từ đó cho phép giải bài toán nhận diện bằng các phương pháp so khớp vector như k-NN

hoặc các thư viện tìm kiếm nhanh. Ý tưởng embedding của FaceNet đã đặt nền móng cho nhiều mô hình nhận diện khuôn mặt hiện đại sau này.

1.4.2. ArcFace / InsightFace

ArcFace (CVPR 2019) cải tiến các mô hình embedding trước đó bằng cách đưa vào Additive Angular Margin Loss, giúp tăng biên phân tách giữa các danh tính trong không gian đặc trưng. Nhờ đó, embedding thu được có khả năng phân biệt cao và ổn định hơn trong các điều kiện thực tế.

InsightFace là dự án mã nguồn mở hiện thực ArcFace và cung cấp nhiều mô hình đã được huấn luyện sẵn, đạt độ chính xác cao trên các bộ dữ liệu chuẩn. Với khả năng tối ưu cho cả CPU và GPU, ArcFace/InsightFace hiện được xem là một trong những chuẩn tham chiếu phổ biến trong các hệ thống Face ID, bao gồm cả bài toán định danh bệnh nhân trong môi trường y tế.

1.4.3. MobileFaceNet và mô hình nhẹ cho thiết bị nhúng

Đối với các hệ thống triển khai trên thiết bị biên hoặc kiosk có tài nguyên hạn chế, các mô hình lớn không phải lúc nào cũng phù hợp. MobileFaceNet là kiến trúc nhẹ, kế thừa từ MobileNet, nhằm giảm số tham số và chi phí tính toán nhưng vẫn giữ được độ chính xác ở mức chấp nhận được.

Khi kết hợp với các hàm mất mát như ArcFace hoặc CosFace, MobileFaceNet có thể đáp ứng yêu cầu nhận diện thời gian thực trên thiết bị cấu hình thấp. Do đó, mô hình này phù hợp cho các ứng dụng Face ID chi phí thấp và có tiềm năng triển khai trong bệnh viện quy mô vừa và nhỏ.

1.5. Các nghiên cứu và hệ thống tương tự

1.5.1. Các công trình trong và ngoài nước về Face ID trong y tế

Trên thế giới, công nghệ nhận diện khuôn mặt đã được nghiên cứu và ứng dụng trong nhiều bài toán của lĩnh vực y tế, đặc biệt là định danh bệnh nhân và kiểm soát truy cập. Các nghiên cứu cho thấy Face ID giúp nâng cao độ chính xác định danh và giảm phụ thuộc vào giấy tờ truyền thống.

Tại Việt Nam, việc ứng dụng Face ID trong y tế hiện vẫn chủ yếu ở mức thử nghiệm hoặc sử dụng cho các mục đích đơn lẻ, chưa được tích hợp sâu vào quy trình tiếp nhận và điều phối khám bệnh thông qua các nền tảng API.

1.5.2. Các hệ thống xếp hàng tự động trong bệnh viện

Các hệ thống xếp hàng hiện nay thường bao gồm kiosk bốc số, bảng hiển thị và phần mềm gọi số. Tuy nhiên, nhiều hệ thống vẫn hoạt động độc lập, chưa gắn kết chặt chẽ với cơ chế định danh bệnh nhân và chưa hỗ trợ điều phối linh hoạt theo thời gian thực.

1.5.3. Nhận xét, khoảng trống nghiên cứu và lý do chọn giải pháp Face ID + bốc số tự động

Từ các nghiên cứu và hệ thống hiện có, có thể nhận thấy vẫn tồn tại khoảng trống trong việc xây dựng một lớp API thống nhất để tích hợp nhận diện khuôn mặt với hệ thống xếp hàng và điều phối khám bệnh theo thời gian thực. Phần lớn các giải pháp hiện nay mới chỉ giải quyết từng bài toán riêng lẻ, chưa hình thành một nền tảng tích hợp linh hoạt.

Xuất phát từ thực tế đó, đề tài lựa chọn nghiên cứu và xây dựng API cho hệ thống định danh bệnh nhân bằng nhận diện khuôn mặt kết hợp điều phối hàng đợi khám bệnh thời gian thực, nhằm khắc phục các hạn chế của mô hình truyền thống và tạo nền tảng cho các hệ thống y tế thông minh trong tương lai.

CHƯƠNG 2: PHÂN TÍCH BÀI TOÁN VÀ THIẾT KẾ API

2.1. Bối cảnh và hai bài toán cần giải quyết

2.1.1. Mô tả bối cảnh tại quầy tiếp nhận

Tại nhiều cơ sở y tế hiện nay, đặc biệt là các bệnh viện công lập và bệnh viện tuyến tỉnh/huyện, quy trình tiếp nhận bệnh nhân vẫn chủ yếu dựa trên hồ sơ giấy và thao tác nhập liệu thủ công. Người bệnh khi đến khám thường phải thực hiện các bước như lấy số thứ tự, chờ đến lượt, xuất trình giấy tờ tùy thân và cung cấp thông tin hành chính để nhân viên tiếp nhận đăng ký dịch vụ khám bệnh.

Trong các khung giờ cao điểm hoặc những thời điểm số lượng bệnh nhân tăng đột biến do yếu tố thời tiết hoặc dịch bệnh, số lượng nhân viên tiếp nhận thường không đáp ứng kịp nhu cầu thực tế. Điều này dẫn đến tình trạng quá tải tại quầy tiếp nhận, thời gian chờ kéo dài và làm giảm đáng kể trải nghiệm của người bệnh.

Sau khi hoàn tất bước tiếp nhận ban đầu, bệnh nhân tiếp tục được hướng dẫn đến các khu vực như phòng khám chuyên khoa, khu xét nghiệm hoặc chẩn đoán hình ảnh, đồng thời phải mang theo các phiếu giấy tương ứng. Khi lượng bệnh nhân dồn vào một số phòng hoặc một bác sĩ trong cùng thời điểm, việc thiếu cơ chế điều phối tự động khiến một số khu vực rơi vào tình trạng quá tải, trong khi các khu vực khác chưa được khai thác hết công suất. Nhân viên y tế vì vậy phải dành nhiều thời gian cho việc gọi số, điều phối và kiểm tra danh sách, thay vì tập trung vào hoạt động chuyên môn.

Bên cạnh đó, tại quầy tiếp nhận, nhân viên phải tra cứu hồ sơ bệnh nhân thông qua nhiều nguồn khác nhau để xác định bệnh nhân mới hay tái khám, tìm lại hồ sơ cũ và gợi ý dịch vụ phù hợp. Với lượng dữ liệu lớn và phân tán ở nhiều hệ thống, quá trình này vừa tốn thời gian, vừa tiềm ẩn nguy cơ sai sót hoặc nhầm lẫn thông tin.

Từ thực tế trên, có thể xác định hai điểm nghẽn chính trong quy trình tiếp nhận và khám bệnh hiện nay:

- Việc định danh và truy xuất hồ sơ bệnh nhân còn mang tính thủ công, phụ thuộc nhiều vào giấy tờ và thao tác của nhân viên, dễ phát sinh chậm trễ và sai sót.
- Công tác bốc số và điều phối bệnh nhân chưa được tự động hóa, chủ yếu dựa vào kinh nghiệm và quan sát thủ công, thiếu công cụ hỗ trợ tối ưu hóa hàng đợi và phân bổ bệnh nhân giữa các phòng khám.

Đây chính là hai bài toán trọng tâm mà hệ thống Face ID kết hợp bốc số tự động trong đề tài hướng tới giải quyết.

2.1.2. Bài toán 1: Định danh và quản lý hồ sơ bệnh nhân bằng Face ID

Để cải thiện hiệu quả định danh bệnh nhân, đề tài đề xuất ứng dụng công nghệ nhận diện khuôn mặt (Face ID) như một phương thức định danh sinh trắc học tại quầy tiếp nhận hoặc kiosk tự phục vụ. Thông qua camera, hệ thống thu nhận ảnh khuôn mặt, trích xuất đặc trưng và thực hiện so khớp với cơ sở dữ liệu đã được xây dựng trước đó để xác định danh tính bệnh nhân.

Trong mô hình nghiên cứu, hệ thống Face ID chỉ đảm nhiệm chức năng xử lý nhận diện và quản lý embedding khuôn mặt, trong khi toàn bộ dữ liệu y tế và thông tin hành chính chi tiết vẫn được quản lý bởi hệ thống HIS/EHR hiện hữu. Việc giao tiếp giữa hai hệ thống được thực hiện thông qua các API, giúp Face ID hoạt động như một lớp hỗ trợ độc lập, không làm thay đổi kiến trúc lõi của hệ thống bệnh viện.

Mục tiêu của bài toán định danh bao gồm:

- Rút ngắn thời gian xác định danh tính bệnh nhân trong khâu tiếp nhận.
- Hạn chế sai sót trong truy xuất hồ sơ, đặc biệt với các trường hợp trùng thông tin hành chính.
- Đảm bảo an toàn và bảo mật dữ liệu khi chỉ lưu trữ vector đặc trưng khuôn mặt thay vì ảnh gốc hay dữ liệu y tế nhạy cảm.

Phạm vi nghiên cứu của bài toán này tập trung vào khía cạnh định danh và liên kết mã bệnh nhân, không đi sâu vào quản lý nội dung hồ sơ bệnh án.

2.1.3. Bài toán 2: Bốc số khám bệnh và quản lý hàng đợi tự động

Song song với bài toán định danh, đề tài nghiên cứu xây dựng một hệ thống bốc số và quản lý hàng đợi khám bệnh tự động. Sau khi được định danh hoặc đăng ký mới, bệnh nhân lựa chọn dịch vụ khám trên kiosk, thông tin này được gửi về máy chủ để thực hiện cấp số thứ tự và phân bổ vào hàng đợi phù hợp.

Hệ thống điều phối được thiết kế nhằm:

- Tự động cấp số thứ tự theo từng dịch vụ và chuyên khoa.
- Phân bổ bệnh nhân hợp lý giữa các phòng khám hoặc bác sĩ có cùng chức năng.
- Cung cấp thông tin hàng đợi theo thời gian thực cho nhân viên điều phối và bác sĩ.

Việc tự động hóa quá trình này giúp giảm áp lực cho khu vực tiếp nhận, tăng tính minh bạch cho người bệnh và nâng cao hiệu quả sử dụng nguồn lực y tế.

2.2. Phân tích bài toán 1 – API nhận diện gương mặt

2.2.1. Các tác nhân liên quan

Bài toán nhận diện gương mặt trong hệ thống có sự tham gia của nhiều tác nhân với vai trò và mức độ tương tác khác nhau, bao gồm: bệnh nhân, nhân viên tiếp nhận, bác sĩ, hệ thống Face ID và hệ thống quản lý hồ sơ y tế HIS/EHR.

Trong đó, bệnh nhân là đối tượng được nhận diện thông qua dữ liệu sinh trắc học khuôn mặt. Nhân viên tiếp nhận và bác sĩ là các tác nhân gián tiếp, sử dụng kết quả định danh để truy xuất và làm việc với hồ sơ bệnh nhân. Hệ thống Face ID đóng vai trò trung tâm trong bài toán này, chịu trách nhiệm tiếp nhận ảnh khuôn mặt, thực hiện tiền xử lý, trích xuất đặc trưng (embedding) và so khớp với cơ sở dữ liệu nhận diện thông qua các API được thiết kế sẵn.

Hệ thống HIS/EHR chịu trách nhiệm quản lý toàn bộ dữ liệu y tế và hành chính của bệnh nhân. Hai hệ thống Face ID và HIS/EHR được thiết kế hoạt động độc lập nhưng có khả năng liên thông thông qua API. Nguyên tắc thiết kế cốt lõi là tách biệt xử lý nhận diện sinh trắc học và dữ liệu y tế, trong đó Face ID chỉ trao đổi các thông tin tối thiểu cần thiết như mã bệnh nhân hoặc trạng thái nhận diện, nhằm đảm bảo an toàn, bảo mật và giảm thiểu rủi ro rò rỉ dữ liệu nhạy cảm.

2.2.2. Các trường hợp sử dụng chính

API nhận diện gương mặt trong hệ thống được xây dựng nhằm hỗ trợ hai trường hợp sử dụng chính.

Trường hợp thứ nhất là đăng ký bệnh nhân mới, trong đó hệ thống tiếp nhận ảnh khuôn mặt từ thiết bị đầu cuối, trích xuất embedding và lưu trữ embedding này gắn với một mã bệnh nhân mới được sinh ra từ hệ thống quản lý hồ sơ. Trường hợp này thường được thực hiện khi bệnh nhân đến khám lần đầu và chưa tồn tại dữ liệu nhận diện trong hệ thống.

Trường hợp thứ hai là nhận diện bệnh nhân tái khám, khi bệnh nhân đứng trước camera tại kiosk hoặc quầy tiếp nhận. Hệ thống Face ID thực hiện so khớp embedding khuôn mặt thu được với cơ sở dữ liệu hiện có và trả về mã bệnh nhân tương ứng nếu tìm thấy kết quả phù hợp. Trong trường hợp độ tin cậy của kết quả nhận diện không đạt ngưỡng cho phép, hệ thống không tự động xác nhận danh tính mà chuyển sang quy trình xác thực thủ công do nhân viên tiếp nhận thực hiện, nhằm đảm bảo an toàn và tránh nhầm lẫn hồ sơ.

2.2.3. Yêu cầu chức năng và phi chức năng

Về yêu cầu chức năng, API nhận diện gương mặt cần hỗ trợ các chức năng cơ bản bao gồm: đăng ký dữ liệu khuôn mặt mới, nhận diện khuôn mặt từ ảnh đầu vào, cập nhật hoặc bổ sung embedding khi cần thiết, và cung cấp các API cho các ứng dụng

phía client như kiosk, quầy tiếp nhận và các hệ thống liên thông khác. Các API này phải được thiết kế thống nhất, rõ ràng và dễ tích hợp nhằm phục vụ cho nhiều kịch bản sử dụng khác nhau trong hệ thống.

Bên cạnh đó, hệ thống cũng phải đáp ứng các yêu cầu phi chức năng quan trọng. Trước hết, độ chính xác nhận diện cần đạt mức phù hợp với môi trường bệnh viện, hạn chế tối đa các trường hợp nhận diện sai. Thời gian phản hồi của API phải đủ nhanh để đáp ứng yêu cầu vận hành thực tế tại quầy tiếp nhận và kiosk tự phục vụ. Ngoài ra, hệ thống cần đảm bảo các yêu cầu về bảo mật dữ liệu, chỉ lưu trữ và xử lý embedding khuôn mặt thay vì ảnh gốc, đồng thời có khả năng mở rộng khi số lượng bệnh nhân và yêu cầu truy cập tăng lên trong các khung giờ cao điểm.

2.3. Phân tích bài toán 2 – Bốc số & quản lý hàng đợi

2.3.1. Các tác nhân liên quan

Trong bài toán bốc số và quản lý hàng đợi, bệnh nhân là tác nhân chính, thực hiện đăng ký dịch vụ tại kiosk, nhận số thứ tự và theo dõi lượt khám của mình thông qua màn hình hiển thị. Nhân viên điều phối hoặc quầy tiếp đón có nhiệm vụ giám sát tổng thể tình trạng hàng đợi, xử lý các trường hợp ưu tiên hoặc ngoại lệ phát sinh trong quá trình tiếp nhận. Bác sĩ và điều dưỡng tại phòng khám sử dụng hệ thống để xem danh sách bệnh nhân đang chờ, đồng thời cập nhật trạng thái các lượt khám như "đang khám" hoặc "đã khám xong".

Hệ thống gọi số chịu trách nhiệm hiển thị và thông báo lượt khám tại các khu vực chờ và trong phòng khám, đảm bảo bệnh nhân được gọi đúng thứ tự. Các màn hình hiển thị và thiết bị client đóng vai trò trình bày trực quan thông tin về thứ tự và trạng thái từng lượt khám cho cả bệnh nhân và nhân viên y tế, góp phần tăng tính minh bạch và giảm áp lực cho khu vực tiếp nhận.

2.3.2. Các ca sử dụng chính

Trong hệ thống bốc số và quản lý hàng đợi, bệnh nhân thực hiện thao tác trên kiosk tự phục vụ để lựa chọn dịch vụ khám và, trong trường hợp cho phép, có thể lựa chọn bác sĩ ưu tiên. Sau khi tiếp nhận yêu cầu, hệ thống tiến hành kiểm tra thông tin đăng ký, phân tích tình trạng tải của từng phòng khám và bác sĩ, từ đó tự động cấp một số thứ tự mới cho bệnh nhân và gán bệnh nhân vào hàng đợi phù hợp. Tại phòng khám, nhân viên y tế hoặc bác sĩ cập nhật trạng thái của từng lượt khám, bao gồm các trạng thái như gọi bệnh nhân vào khám, đang khám và hoàn thành khám. Trên cơ sở đó, hệ thống tiếp tục điều phối bệnh nhân tiếp theo theo đúng thứ tự, đồng thời cập nhật trạng thái hàng đợi theo thời gian thực lên các màn hình hiển thị và dashboard điều phối để phục vụ công tác theo dõi và quản lý.

2.3.3. Yêu cầu chức năng

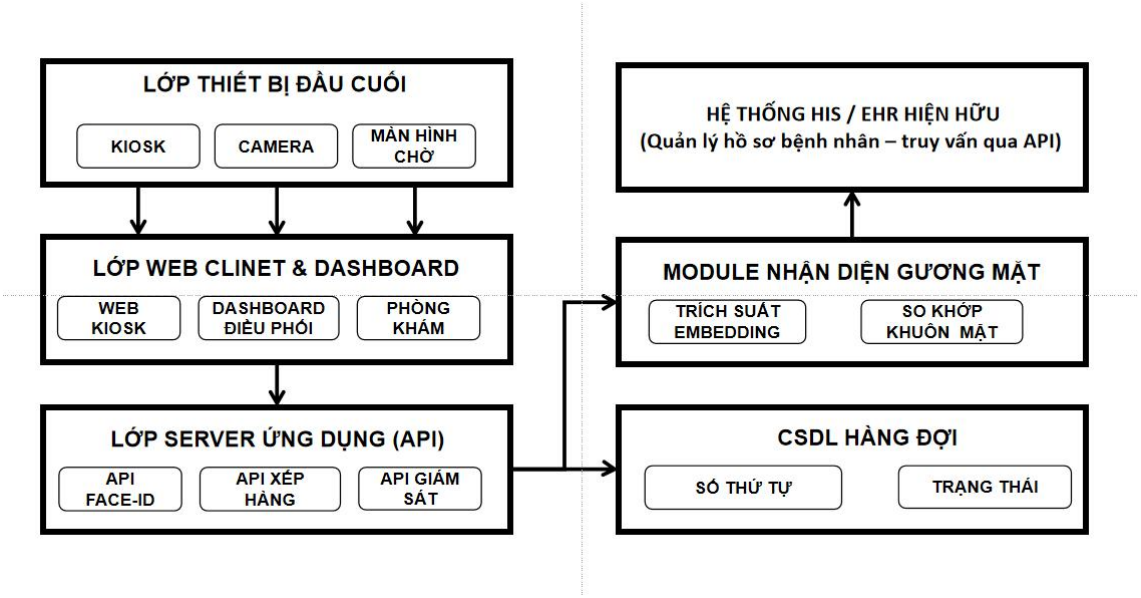
Hệ thống cần đáp ứng các yêu cầu chức năng cơ bản nhằm bảo đảm quá trình bốc số và điều phối bệnh nhân diễn ra chính xác và hiệu quả. Cụ thể, hệ thống phải có khả năng sinh số thứ tự tự động, không trùng lặp và có thể phân loại theo từng khu vực hoặc dịch vụ khám. Đồng thời, hệ thống cho phép quản lý hàng đợi riêng cho từng phòng khám, hỗ trợ các thao tác như thêm, xóa, chuyển lượt khám hoặc ưu tiên xử lý một số trường hợp đặc biệt như cấp cứu hoặc các đối tượng ưu tiên theo quy định. Bên cạnh đó, hệ thống cần tự động lựa chọn phòng khám hoặc bác sĩ phù hợp cho bệnh nhân dựa trên dịch vụ đã đăng ký và các tham số cấu hình của hệ thống, bao gồm chuyên khoa, năng lực tiếp nhận của phòng khám và mức tải hiện tại. Ngoài ra, hệ thống phải cung cấp các giao diện lập trình ứng dụng (API – Application Programming Interface) để các thành phần như phòng khám, quầy tiếp nhận và dashboard điều phối có thể truy cập và cập nhật trạng thái hàng đợi một cách thống nhất.

2.3.4. Yêu cầu phi chức năng

Bên cạnh các yêu cầu chức năng, hệ thống cần đáp ứng các yêu cầu phi chức năng nhằm bảo đảm khả năng vận hành ổn định trong môi trường bệnh viện. Trước hết, hệ thống phải hỗ trợ cập nhật thông tin theo thời gian thực, trong đó dữ liệu gọi số và trạng thái hàng đợi được phản ánh ngay khi có thay đổi với độ trễ nhỏ. Về độ tin cậy, hệ thống cần bảo đảm không làm mất dữ liệu các lượt khám trong trường hợp xảy ra sự cố mạng hoặc mất điện tạm thời, đồng thời có cơ chế khôi phục trạng thái hoạt động khi hệ thống được khởi động lại. Ngoài ra, hệ thống phải có khả năng chịu tải tốt, duy trì hoạt động ổn định ngay cả khi số lượng yêu cầu tăng cao vào các khung giờ cao điểm. Cuối cùng, giao diện người dùng cần được thiết kế rõ ràng, dễ sử dụng đối với cả bệnh nhân và nhân viên y tế, nhằm thuận tiện cho việc triển khai và vận hành trong môi trường bệnh viện thực tế.

2.4. Mô hình kiến trúc tổng thể hệ thống tích hợp

2.4.1. Kiến trúc tổng quan



Hình 2.1 Mô tả kiến trúc tổng quan

Hình 2.1 mô tả kiến trúc tổng quan của hệ thống, được thiết kế theo mô hình phân lớp nhằm đảm bảo tính rõ ràng trong tổ chức, khả năng mở rộng và thuận tiện cho việc tích hợp với các hệ thống hiện hữu của bệnh viện. Ở lớp thiết bị đầu cuối, hệ thống bao gồm các kiosk đặt tại khu vực tiếp nhận, camera bố trí tại quầy và phòng khám phục vụ nhận diện khuôn mặt, cùng các màn hình hiển thị tại khu vực chờ để thông báo số thứ tự và trạng thái khám bệnh cho bệnh nhân. Lớp này đóng vai trò trực tiếp tương tác với người sử dụng, thực hiện thu thập dữ liệu đầu vào và hiển thị kết quả xử lý từ hệ thống.

Lớp server ứng dụng đóng vai trò trung tâm xử lý nghiệp vụ, triển khai các API phục vụ cho mô-đun Face ID, chức năng bốc số và cập nhật trạng thái hàng đợi. Server ứng dụng cũng chịu trách nhiệm cung cấp dữ liệu cho dashboard điều phối và các web client khác trong hệ thống. Song song với đó, mô-đun nhận diện khuôn mặt được triển khai trên server hoặc một máy chuyên dụng riêng, đảm nhiệm việc trích xuất embedding khuôn mặt và thực hiện so khớp với cơ sở dữ liệu nhận diện.

Về lưu trữ dữ liệu, cơ sở dữ liệu hồ sơ bệnh nhân được quản lý bởi hệ thống HIS/EHR hiện hữu của bệnh viện. Server ứng dụng không can thiệp trực tiếp vào dữ liệu này mà chỉ truy vấn thông qua các API liên thông. Bên cạnh đó, hệ thống duy trì một cơ sở dữ liệu hàng đợi riêng để lưu trữ thông tin lượt khám, số thứ tự và trạng thái của từng bệnh nhân trong quá trình xếp hàng và điều phối. Các web client và dashboard cung cấp giao diện cho kiosk, quầy tiếp nhận, phòng khám và bộ phận điều phối, cho phép theo dõi và quản lý hoạt động của toàn bộ hệ thống một cách trực quan.

2.4.2. Các phân hệ chính

Hệ thống được chia thành các phân hệ chức năng chính, phối hợp với nhau để đảm bảo quy trình tiếp nhận và khám bệnh diễn ra liền mạch. Phân hệ Face ID tại quầy tiếp nhận và kiosk cho phép chụp ảnh khuôn mặt bệnh nhân, gửi yêu cầu nhận diện lên server và hỗ trợ cả hai trường hợp đăng ký bệnh nhân mới và nhận diện bệnh nhân tái khám.

Phân hệ nhận diện khuôn mặt trên server chịu trách nhiệm tiền xử lý ảnh, trích xuất embedding và so khớp với cơ sở dữ liệu đã lưu. Kết quả trả về cho các ứng dụng phía trên bao gồm mã bệnh nhân tương ứng hoặc thông báo không tìm thấy kết quả khớp, làm cơ sở cho các bước xử lý tiếp theo.

Phân hệ quản lý hồ sơ bệnh nhân thực hiện việc gắn kết mã bệnh nhân nhận được từ hệ thống Face ID với hồ sơ bệnh án trong HIS/EHR. Nhờ đó, nhân viên y tế và bác sĩ có thể mở nhanh hồ sơ bệnh nhân mà không cần thực hiện các thao tác tìm kiếm thủ công phức tạp.

Phân hệ bốc số và quản lý hàng đợi tiếp nhận yêu cầu dịch vụ từ kiosk, sinh số thứ tự và phân bổ bệnh nhân vào hàng đợi của các phòng khám hoặc bác sĩ phù hợp. Phân hệ này đóng vai trò điều phối luồng bệnh nhân trong toàn bộ quá trình khám chữa bệnh.

Cuối cùng, dashboard điều phối và màn hình gọi số cung cấp giao diện hiển thị trạng thái hoạt động của các phòng khám, số lượng bệnh nhân đang chờ và thứ tự khám. Đồng thời, hệ thống cho phép nhân viên điều phối can thiệp khi cần thiết, chẳng hạn như chuyển phòng, áp dụng ưu tiên hoặc tạm dừng một phòng khám, nhằm đảm bảo hoạt động của bệnh viện diễn ra thông suốt.

2.5. Thiết kế luồng nghiệp vụ chi tiết

2.5.1. Quy trình bệnh nhân mới

Đối với bệnh nhân đến khám lần đầu, quy trình bắt đầu khi bệnh nhân tiếp cận kiosk và lựa chọn chức năng “Đăng ký lần đầu”. Tại đây, bệnh nhân cung cấp một số thông tin hành chính cơ bản theo yêu cầu của hệ thống. Kiosk tiến hành chụp ảnh khuôn mặt, sau đó mô-đun Face ID trích xuất đặc trưng và sinh mã bệnh nhân mới. Thông tin định danh này được chuyển sang hệ thống HIS/EHR để hoàn thiện hồ sơ bệnh nhân ban đầu theo quy trình hiện hành của bệnh viện. Sau khi hồ sơ được tạo, bệnh nhân lựa chọn các dịch vụ khám cần thiết, hệ thống thực hiện bốc số và đưa bệnh nhân vào hàng đợi tương ứng của các phòng khám.

2.5.2. Quy trình bệnh nhân tái khám

Đối với bệnh nhân tái khám, bệnh nhân đến kiosk hoặc quầy tiếp nhận và lựa chọn chức năng “Tái khám”, đồng thời đứng trước camera để hệ thống thực hiện nhận diện

khuôn mặt. Mô-đun Face ID tiến hành so khớp ảnh chụp với cơ sở dữ liệu embedding đã lưu và trả về mã bệnh nhân trong trường hợp tìm thấy kết quả phù hợp. Dựa trên mã bệnh nhân này, hệ thống HIS/EHR tự động mở hồ sơ tương ứng, giúp bệnh nhân và nhân viên tiếp nhận tiếp tục các bước đăng ký dịch vụ khám cho lần khám hiện tại. Cuối cùng, hệ thống bốc số và cập nhật bệnh nhân vào hàng đợi của phòng khám hoặc bác sĩ phù hợp, đảm bảo quy trình tiếp nhận diễn ra nhanh chóng và nhất quán.

2.5.3. Quy trình điều phối bệnh nhân đến phòng khám/bác sĩ

Trong quá trình vận hành, server liên tục theo dõi số lượng bệnh nhân đang chờ tại từng phòng khám cũng như trạng thái làm việc của các bác sĩ. Khi một lượt khám kết thúc, hệ thống tự động ghi nhận sự kiện này và kích hoạt cơ chế điều phối để gọi bệnh nhân tiếp theo theo đúng thứ tự trong hàng đợi đã được xác định. Toàn bộ trạng thái của các phòng khám, thứ tự lượt khám và tình hình hàng chờ được cập nhật theo thời gian thực lên dashboard điều phối và các màn hình hiển thị. Nhờ đó, bệnh nhân có thể dễ dàng theo dõi thứ tự của mình, đồng thời bệnh viện có cơ sở trực quan để đánh giá mức độ tải và phân bổ nguồn lực hợp lý tại từng khu vực khám.

2.6. Thiết kế cơ sở dữ liệu

2.6.1. Thiết kế cấu trúc dữ liệu cho mô-đun "Bốc số"

Dữ liệu của hệ thống gồm các bản sau:

Bảng 2.1. Danh mục dịch vụ

STT	Tên trường dữ liệu	Kiểu dữ liệu	Diễn giải
1	service_id(Primary)	String	Mã dịch vụ (DV001, DV700,…)
2	service_name	String	Tên dịch vụ
3	specialty	String	Chuyên khoa tương ứng (Nội tổng quát, Nhi, Chẩn đoán hình ảnh…).
4	room_id	String	Phòng thực hiện chính được cấu hình mặc định.
5	prioritize	Boolean	Cờ ưu tiên (ví dụ: dịch vụ cấp cứu, dịch vụ ưu tiên).

Bảng 2.2. Danh mục phòng khám

STT	Tên trường dữ liệu	Kiểu dữ liệu	Diễn giải
1	room_id (Primary)	String	Mã phòng (P001, P002, P601,…).
2	room_name	String	Tên phòng (Phòng khám Nội tổng quát, Phòng Siêu âm…).
3	room_type	String	Loại phòng (khám, xét nghiệm, chẩn đoán hình ảnh…).
4	active	Boolean	Trạng thái sử dụng (1: đang hoạt động, 0: tạm ngưng).

Bảng 2.3. Danh mục bác sĩ

STT	Tên trường dữ liệu	Kiểu dữ liệu	Diễn giải
1	doctor_id	String	Mã bác sĩ (DR000001,…)
2	doctor's_name	String	Họ tên bác sĩ
3	specialty	String	Chuyên khoa/vai trò
4	room_id	String	Phòng khám chính mà bác sĩ thường làm việc.

Bảng 2.4. Hàng đợi tạm thời trên hệ thống

STT	Tên trường dữ liệu	Kiểu dữ liệu	Diễn giải
1	wait_index	String	Mã dòng hàng đợi đại diện cho 1 vị trí tạm thời đang xếp hàng chờ.
2	P001	String	Tên các phòng của bệnh viện phòng khám,sau khi kiosk gửi dịch vụ được yêu cầu về và được hệ thống phân loại thì thì số thứ tự lượt khám của bệnh nhân sẽ được lưu tạm ở đây để điều phố sau.Nguyên tắt mỗi dòng là 1 vị ví đang chời lên 1 mã bệnh nhân chĩ được xuất hiện ở 1 cột duy nhất trong mỗi dòng.
3	P002	String	
4	…	…	

Bảng 2.5. Trạng thái thời gian thực của phòng

STT	Tên trường dữ liệu	Kiểu dữ liệu	Diễn giải
1	real_time_index	String	Mã dòng hàng đợi đại diện cho 1 vị trí đang xếp hàng chờ thực tế trước mỗi phòng để chuẩn bị vào khám.
2	P001	String	Tương tự wait_queue mỗi cột điều đại diện cho 1 phòng khám và số thứ tự khám củ bệnh nhân sẽ xếp vào dây,nhưng khác đây và hàng xếp thực tế theo thời gian thực còn wait_queue là hàng chờ tạm thời để điều phối.về nguyên tắc thì thương tự wait_queue.
3	P002	String	
4	…	…	

Bảng 2.6. Nhật ký tiếp nhận

STT	Tên trường dữ liệu	Kiểu dữ liệu	Diễn giải
1	patient_id	String	Sô thứ tự lượt khám trong ngày được xếp tự dộng tăng
2	receptio_ time	String	Thời điểm tiếp nhận
3	service	String	Lưu trữ các dịch vụ mà bệnh nhân đã chọn kèm bác sĩ chuẩn đoán.
4	status	Boolean	Trạng thái của lượt khám đã hoàn thành chưa được sử dụng để là cờ cho các client theo dõi điều phối.

2.6.2. Thiết kế cấu trúc dữ liệu cho mô-đun Face ID

Mô-đun Face ID trong hệ thống được thiết kế như một thành phần xử lý độc lập, triển khai trên một server API phụ nhằm đảm bảo hiệu năng và khả năng mở rộng. Khác với dữ liệu nghiệp vụ của module bốc số được lưu trữ trong các bảng quan hệ MySQL, dữ liệu nhận diện khuôn mặt được tổ chức dưới dạng cơ sở dữ liệu vector và lưu trong các tệp nhị phân. Cách tiếp cận này giúp tối ưu tốc độ truy vấn cũng như giảm dung lượng lưu trữ khi số lượng bệnh nhân tăng lên.

Trong mô hình đề xuất, mỗi bệnh nhân được biểu diễn bởi một vector đặc trưng khuôn mặt (embedding) có kích thước 512 chiều, liên kết trực tiếp với mã bệnh nhân (patient_id). Embedding này được sử dụng làm đại diện định danh để thực hiện so khớp khuôn mặt trong quá trình xác thực bệnh nhân.

2.6.2.1. Cấu trúc dữ liệu chính của mô-đun Face ID

Dữ liệu nhận diện khuôn mặt được quản lý tập trung trong thư mục database của mô-đun facial_recognition. Trong đó, ảnh gốc của bệnh nhân được lưu trong thư mục database/image, mỗi ảnh tương ứng với một bệnh nhân và tên tệp được dùng làm nhãn định danh cho embedding. Các embedding được trích xuất từ ảnh khuôn mặt được lưu trữ trong tệp facial_recognition.npz, bao gồm ma trận embedding, danh sách nhãn bệnh nhân và đường dẫn đến ảnh gốc tương ứng.

Để tăng tốc tìm kiếm, toàn bộ embedding được xây dựng thành một chỉ mục FAISS và lưu trong tệp nhị phân facial_recognition.faiss. Chỉ mục này cho phép thực hiện tìm kiếm các vector gần nhất với độ trễ thấp, phù hợp cho hệ thống nhận diện thời gian thực. Ngoài ra, hệ thống sử dụng thêm một tệp cache embedding nhằm tránh việc trích xuất lại đặc trưng cho các ảnh đã xử lý trước đó, giúp rút ngắn thời gian khởi tạo và cập nhật cơ sở dữ liệu.

Khi hệ thống khởi động, mô-đun nhận diện sẽ nạp chỉ mục FAISS và cơ sở dữ liệu embedding vào bộ nhớ. Trong trường hợp có GPU, chỉ mục có thể được chuyển sang

GPU để tăng tốc truy vấn. Pipeline xử lý ảnh bao gồm các bước phát hiện khuôn mặt, cắt vùng mặt và trích xuất embedding đã chuẩn hóa bằng mô hình InsightFace.

2.6.2.2. Trong triển khai thực tế

Dữ liệu nhận diện khuôn mặt được quản lý tập trung trong thư mục database của mô-đun facial_recognition. Trong đó, ảnh gốc của bệnh nhân được lưu trong thư mục database/image, mỗi ảnh tương ứng với một bệnh nhân và tên tệp được dùng làm nhãn định danh cho embedding. Các embedding được trích xuất từ ảnh khuôn mặt được lưu trữ trong tệp facial_recognition.npz, bao gồm ma trận embedding, danh sách nhãn bệnh nhân và đường dẫn đến ảnh gốc tương ứng.

Để tăng tốc tìm kiếm, toàn bộ embedding được xây dựng thành một chỉ mục FAISS và lưu trong tệp nhị phân facial_recognition.faiss. Chỉ mục này cho phép thực hiện tìm kiếm các vector gần nhất với độ trễ thấp, phù hợp cho hệ thống nhận diện thời gian thực. Ngoài ra, hệ thống sử dụng thêm một tệp cache embedding nhằm tránh việc trích xuất lại đặc trưng cho các ảnh đã xử lý trước đó, giúp rút ngắn thời gian khởi tạo và cập nhật cơ sở dữ liệu.

Khi hệ thống khởi động, mô-đun nhận diện sẽ nạp chỉ mục FAISS và cơ sở dữ liệu embedding vào bộ nhớ. Trong trường hợp có GPU, chỉ mục có thể được chuyển sang GPU để tăng tốc truy vấn. Pipeline xử lý ảnh bao gồm các bước phát hiện khuôn mặt, cắt vùng mặt và trích xuất embedding đã chuẩn hóa bằng mô hình InsightFace.

2.7. Thiết kế giao tiếp giữa thiết bị và máy chủ

Các thiết bị đầu cuối trong hệ thống, bao gồm kiosk, camera tại quầy tiếp nhận và phòng khám, màn hình hiển thị số cũng như dashboard điều phối, giao tiếp với máy chủ thông qua hai cơ chế chính. Cơ chế thứ nhất là API RESTful, phục vụ các thao tác đồng bộ như gửi ảnh nhận diện, đăng ký dịch vụ, bốc số và cập nhật trạng thái hàng đợi. Cơ chế thứ hai là kênh truyền thông thời gian thực, sử dụng WebSocket hoặc Server-Sent Events (SSE), nhằm truyền các sự kiện cập nhật hàng đợi và gọi số đến các client mà không cần tải lại trang.

2.7.1. API cho thiết bị Face ID & kiosk

Hệ thống cung cấp các nhóm API phục vụ cho mô-đun Face ID, kiosk đăng ký dịch vụ và các phòng khám. Các API Face ID cho phép đăng ký khuôn mặt mới và nhận diện bệnh nhân từ ảnh đầu vào. Nhóm API dành cho kiosk hỗ trợ đăng ký khám cho bệnh nhân mới hoặc tái khám, tạo lượt khám và đưa bệnh nhân vào hàng đợi phù hợp. Nhóm API cho phòng khám và nhân viên điều phối cho phép truy vấn danh sách bệnh nhân đang chờ, gọi bệnh nhân tiếp theo và cập nhật trạng thái hoàn thành lượt khám. Các API được tổ chức theo từng router chức năng, giúp cấu trúc mã nguồn rõ ràng và thuận tiện cho việc mở rộng, bảo trì.

2.7.2. Cơ chế realtime (WebSocket, SSE…) cho dashboard và màn hình gọi số

Để đảm bảo thông tin hiển thị luôn đồng bộ và cập nhật kịp thời, hệ thống sử dụng WebSocket hoặc SSE làm kênh truyền thông thời gian thực. Khi client kết nối đến máy chủ, mọi thay đổi quan trọng trong hàng đợi như thêm bệnh nhân mới, gọi bệnh nhân hoặc kết thúc lượt khám sẽ được server gửi dưới dạng thông điệp sự kiện. Client nhận sự kiện và cập nhật giao diện ngay lập tức, giúp dashboard điều phối và màn hình gọi số luôn phản ánh đúng trạng thái thực tế của hệ thống.

Nhờ sự kết hợp giữa API RESTful và kênh truyền thông thời gian thực, hệ thống Face ID kết hợp bốc số tự động đáp ứng tốt yêu cầu cập nhật liên tục trong môi trường bệnh viện, đồng thời vẫn giữ được kiến trúc rõ ràng và khả năng tích hợp với các hệ thống HIS/EHR hiện hữu.

CHƯƠNG 3: XÂY DỰNG VÀ TRIỂN KHAI API

3.1. Môi trường và công nghệ sử dụng

Hệ thống được xây dựng dựa trên kiến trúc client–server, trong đó máy chủ chịu trách nhiệm xử lý nhận diện khuôn mặt, bốc số – điều phối khám bệnh, còn ứng dụng web cung cấp giao diện cho bệnh nhân, lễ tân và bác sĩ. Các công nghệ được lựa chọn nhằm đảm bảo hiệu năng, khả năng mở rộng và dễ triển khai trong thực tế.

3.1.1. Server: Python FastAPI, CSDL MySQL/SQLite

Backend của hệ thống được phát triển bằng Python FastAPI, một framework hiện đại có hiệu năng cao, hỗ trợ mô hình lập trình bất đồng bộ và tích hợp tốt với WebSocket – phù hợp để xây dựng dashboard thời gian thực. FastAPI cho phép tách các chức năng thành nhiều router độc lập như: nhận diện khuôn mặt, đăng ký dịch vụ, điều phối hàng đợi, và quản lý thông tin phòng khám – bác sĩ.

Dữ liệu của hệ thống được lưu trữ trong MySQL 8.0, bao gồm danh mục dịch vụ, phòng khám, bác sĩ, hàng đợi tạm thời (wait_queue), hàng đợi thời gian thực (real_time) và nhật ký tiếp nhận (reception_log). MySQL được lựa chọn vì độ ổn định, tốc độ truy vấn cao và phù hợp triển khai lâu dài.
Ngoài ra, SQLite cũng được sử dụng trong giai đoạn thử nghiệm hoặc chạy bộ nhỏ lẻ, do tính đơn giản và không cần cấu hình.

3.1.2. Ứng dụng web: Vue3 (Kiosk, Dashboard, trang quản trị)

Giao diện người dùng của hệ thống được xây dựng bằng Vue3 – một framework web hiện đại, nhẹ và linh hoạt, phù hợp cho việc phát triển các ứng dụng có yêu cầu cập nhật thời gian thực và khả năng mở rộng cao. Toàn bộ ứng dụng web được chia thành ba phân hệ chính, tương ứng với các nhóm người dùng khác nhau trong môi trường bệnh viện.

Phân hệ thứ nhất là giao diện Kiosk, phục vụ trực tiếp cho bệnh nhân trong quá trình đăng ký khám bệnh. Thông qua giao diện này, bệnh nhân có thể lựa chọn dịch vụ khám, đăng ký lượt khám và xác thực danh tính bằng Face ID hoặc nhập thông tin thủ công khi cần thiết. Giao diện Kiosk được thiết kế tối ưu cho màn hình lớn và thiết bị cảm ứng, giúp thao tác nhanh, dễ hiểu và hạn chế sai sót trong quá trình sử dụng.

Phân hệ thứ hai là Dashboard điều phối, dành cho nhân viên lễ tân và bộ phận điều phối theo dõi tình trạng hàng đợi và hoạt động của các phòng khám. Dashboard hiển thị các thông tin như số thứ tự đang được gọi, số tiếp theo và danh sách bệnh nhân đang chờ tại từng phòng. Dữ liệu trên Dashboard được cập nhật theo thời gian thực thông qua WebSocket, đảm bảo thông tin luôn đồng bộ và phản ánh chính xác trạng thái hiện tại của hệ thống.

Phân hệ thứ ba là trang quản trị, dành cho quản trị viên hệ thống. Trang này cho phép cấu hình các danh mục như dịch vụ khám bệnh, phòng khám, bác sĩ và truy xuất các báo cáo thống kê liên quan đến hoạt động tiếp nhận và khám bệnh. Việc cung cấp giao diện quản trị giúp giảm phụ thuộc vào thao tác trực tiếp trên cơ sở dữ liệu, đồng thời nâng cao tính an toàn và thuận tiện trong quản lý hệ thống.

Tổng thể, hệ thống được xây dựng trên kiến trúc kết hợp FastAPI ở phía backend để xử lý nghiệp vụ và kết nối cơ sở dữ liệu, MySQL hoặc SQLite để lưu trữ dữ liệu bệnh nhân và hàng đợi, cùng với Vue3 ở phía frontend để triển khai các giao diện Kiosk, Dashboard và trang quản trị. Sự kết hợp này tạo thành một nền tảng linh hoạt, hiệu quả và dễ triển khai, đáp ứng tốt yêu cầu vận hành trong môi trường bệnh viện.

3.2. Triển khai module nhận diện khuôn mặt

Mô-đun nhận diện khuôn mặt được xây dựng nhằm hỗ trợ quản lý hồ sơ bệnh nhân dựa trên công nghệ sinh trắc học, cung cấp khả năng xác định danh tính nhanh chóng và tự động. Pipeline xử lý bao gồm ba bước chính: phát hiện khuôn mặt (Detection), trích xuất đặc trưng (Embedding) và đối sánh – ra quyết định (Matching). Mô-đun sử dụng YOLO-face để phát hiện khuôn mặt và áp dụng cơ chế lọc hai lớp (two-stage filtering) để cân bằng tốc độ và độ chính xác khi cơ sở dữ liệu mở rộng.

Hệ thống được triển khai dưới dạng dịch vụ tách biệt và giao tiếp thông qua API. Dữ liệu nhạy cảm như embedding và thông tin danh tính được lưu trữ tại hệ thống trung tâm của bệnh viện, đảm bảo yêu cầu bảo mật và kiểm soát truy cập. Server Face ID chỉ xử lý các yêu cầu nhận diện như hàm *check_face_in_db()* và trả về kết luận tương ứng.

3.2.1. Chuẩn bị dữ liệu và tiền xử lý ảnh

3.2.1.1. Thu thập và tổ chức dữ liệu

Nguồn dữ liệu sử dụng trong giai đoạn phát triển được lấy từ tập dữ liệu công khai trên nền tảng Kaggle. Sau khi lọc và tách dữ liệu, thu được 10.473 ảnh, tương ứng với 10.473 danh tính khác nhau. Dữ liệu được tổ chức theo nguyên tắc mỗi người – một ảnh duy nhất, nhằm mô phỏng điều kiện vận hành thực tế trong bệnh viện, nơi quy trình tiếp nhận bệnh nhân cần đơn giản, chi phí thấp và dễ mở rộng.

Mỗi ảnh được đặt tên theo mã bệnh nhân (ví dụ: 000000.png, Nguyen_Van_A.png) và được lưu trữ trong một thư mục ảnh gốc. Tên file (phần stem) được sử dụng làm nhãn liên kết giữa ảnh và hồ sơ bệnh nhân trong cơ sở dữ liệu embedding.

Trong quá trình xây dựng database, chương trình quét toàn bộ thư mục ảnh, kiểm tra định dạng hợp lệ (JPG, PNG…), đọc ảnh bằng OpenCV và tổng hợp danh sách đường dẫn, tên bệnh nhân và dữ liệu embedding để phục vụ các bước xử lý tiếp theo.

3.2.1.2. Quy trình tiền xử lý

Ttrước khi đưa ảnh vào mô-đun Face ID, hệ thống áp dụng các bước tiền xử lý thống nhất nhằm đảm bảo chất lượng và tính ổn định của embedding:

- Giảm kích thước theo cạnh dài (≤ 512 hoặc ≤ 1024 tuỳ cấu hình), giữ nguyên tỉ lệ gốc nhằm tối ưu tốc độ tính toán nhưng vẫn đảm bảo đủ thông tin khuôn mặt.
- Chuyển đổi không gian màu từ BGR sang RGB để tương thích với các mô hình học sâu được huấn luyện trên định dạng RGB.
- Phát hiện khuôn mặt lần đầu bằng Dlib CNN hoặc YOLO-face, sau đó chọn khuôn mặt lớn nhất làm đối tượng nhận diện nhằm tránh tối đa ảnh hưởng của nền và sẽ đảm bảo có đặt trưng gương mặt trong ảnh.

Các bước tiền xử lý này được thiết kế để đảm bảo tính nhất quán giữa ảnh trong cơ sở dữ liệu và ảnh truy vấn, hạn chế các trường hợp ảnh quá nhỏ, mờ hoặc lệch khung làm suy giảm chất lượng đặc trưng.

3.2.2. Kiến trúc mô-đun Face ID (Detection – Embedding – Matching)

Mô-đun Face ID của hệ thống được thiết kế theo kiến trúc pipeline ba khối liên tiếp, bao gồm phát hiện khuôn mặt (Face Detection), trích xuất đặc trưng (Face Embedding) và so khớp – ra quyết định (Face Matching & Decision). Trong kiến trúc này, ảnh đầu vào lần lượt đi qua từng khối xử lý, từ việc xác định vị trí khuôn mặt, chuyển đổi khuôn mặt thành vector đặc trưng cho đến bước so sánh với cơ sở dữ liệu để đưa ra kết luận định danh. Cách thiết kế theo dạng pipeline giúp mô-đun có cấu trúc rõ ràng, dễ bảo trì, cho phép thay thế hoặc nâng cấp từng thành phần một cách độc lập mà không ảnh hưởng đến toàn bộ hệ thống, đồng thời đảm bảo hiệu năng ổn định khi vận hành.

3.2.2.1. Khối Detection

Khối Detection có nhiệm vụ xác định vị trí khuôn mặt trong ảnh đầu vào và cắt ra vùng chứa khuôn mặt đã được chuẩn hóa trước khi chuyển sang khối Embedding. Trong quá trình nghiên cứu, luận văn triển khai và so sánh hai phương pháp phát hiện khuôn mặt nhằm đánh giá độ chính xác, tốc độ và tính phù hợp với môi trường bệnh viện.

Phương pháp thứ nhất sử dụng các bộ phát hiện khuôn mặt của thư viện Dlib, bao gồm bộ phát hiện dựa trên HOG (Histogram of Oriented Gradients) và bộ phát hiện dựa trên mạng nơ-ron tích chập (CNN) đã được huấn luyện sẵn. Quy trình xử lý bao gồm việc đọc ảnh đầu vào, chuyển đổi sang không gian màu RGB, phát hiện khuôn mặt, lựa chọn bounding box phù hợp và cắt vùng mặt để đưa sang khối Embedding. Cách tiếp cận này có ưu điểm là triển khai đơn giản, ít phụ thuộc vào mô hình bên

ngoài và hoạt động tương đối tốt trên CPU, đặc biệt với bộ phát hiện HOG. Tuy nhiên, trong các thử nghiệm thực tế, phương pháp Dlib bộc lộ nhiều hạn chế khi khuôn mặt bị nghiêng, che khuất hoặc trong điều kiện ánh sáng phức tạp. Phiên bản CNN tuy có độ chính xác cao hơn nhưng tốc độ xử lý chậm trên CPU, làm giảm khả năng đáp ứng thời gian thực. Do đó, Dlib chỉ được sử dụng như một phương án baseline phục vụ nghiên cứu và so sánh, không được lựa chọn cho triển khai chính thức.

Phương pháp thứ hai sử dụng YOLO-face, một biến thể của kiến trúc YOLO được fine-tune riêng cho bài toán phát hiện khuôn mặt. Ảnh đầu vào được chuẩn hóa và resize về kích thước cố định trước khi đưa vào mô hình YOLO-face để suy luận. Mô hình trả về danh sách các bounding box kèm theo độ tin cậy, từ đó hệ thống lựa chọn bounding box có confidence cao nhất, mở rộng thêm một khoảng margin nhất định và cắt ra vùng khuôn mặt để chuyển sang khối Embedding. Kết quả thực nghiệm cho thấy YOLO-face có khả năng phát hiện khuôn mặt ổn định trong nhiều điều kiện khác nhau như khuôn mặt nghiêng, thay đổi ánh sáng hoặc xuất hiện nhiều người trong cùng một khung hình. Đồng thời, phương pháp này đạt tốc độ xử lý cao, đặc biệt khi tận dụng GPU, và dễ dàng nâng cấp lên các phiên bản YOLO mới khi cần thiết.

Từ các phân tích và thử nghiệm trên, YOLO-face được lựa chọn làm phương pháp phát hiện khuôn mặt chính thức trong pipeline của mô-đun Face ID, đáp ứng tốt yêu cầu về độ chính xác, tốc độ và tính ổn định trong môi trường bệnh viện.

3.2.2.2. Khối Embedding – Các phương án trích xuất đặc trưng

Khối Embedding đảm nhiệm vai trò chuyển đổi vùng ảnh khuôn mặt sau bước phát hiện (Detection) thành một vector đặc trưng số đại diện cho danh tính của mỗi bệnh nhân. Chất lượng của embedding ảnh hưởng trực tiếp đến độ chính xác của quá trình so khớp (Matching), đặc biệt trong các hệ thống nhận diện khuôn mặt quy mô lớn và hoạt động trong điều kiện môi trường phức tạp như bệnh viện. Vì vậy, việc lựa chọn mô hình trích xuất đặc trưng cần được đánh giá cẩn trọng dựa trên các tiêu chí như khả năng phân biệt danh tính, độ ổn định trước sự thay đổi ánh sáng và góc mặt, tốc độ suy luận, yêu cầu phần cứng và mức độ tương thích với cơ chế tìm kiếm vector.

Các phương pháp trích xuất đặc trưng truyền thống như LBPH, HOG, SIFT hay SURF chủ yếu dựa trên histogram hoặc gradient của ảnh. Ưu điểm của nhóm phương pháp này là dễ triển khai, không yêu cầu phần cứng mạnh và có thể hoạt động tốt trong các bài toán đơn giản với điều kiện ảnh ổn định. Tuy nhiên, các đặc trưng này rất nhạy với góc quay, ánh sáng và biểu cảm khuôn mặt, dẫn đến độ chính xác thấp trong các tình huống thực tế (in-the-wild). Do đó, các phương pháp truyền thống không phù hợp để áp dụng cho hệ thống nhận diện khuôn mặt trong môi trường bệnh viện, nơi điều kiện chụp ảnh đa dạng và số lượng bệnh nhân lớn.

Nhóm phương pháp học sâu đời đầu, tiêu biểu là FaceNet và Dlib ResNet, đánh dấu bước tiến quan trọng trong lĩnh vực nhận diện khuôn mặt. FaceNet sử dụng hàm mất mát Triplet Loss để học embedding sao cho các vector của cùng một người nằm gần nhau và khác người nằm xa nhau trong không gian đặc trưng. Dlib ResNet, được tích hợp trong thư viện face_recognition, tạo ra embedding 128 chiều và được sử dụng rộng rãi nhờ tính đơn giản và khả năng chạy tốt trên CPU. Mặc dù các mô hình này có ưu điểm là nhẹ, tốc độ suy luận nhanh và dễ triển khai, nhưng khả năng phân biệt danh tính còn hạn chế so với các mô hình hiện đại. Khi điều kiện ảnh khó hơn hoặc khi số lượng người trong cơ sở dữ liệu tăng lên, nguy cơ nhầm lẫn danh tính tăng đáng kể. Vì vậy, nhóm mô hình này chỉ phù hợp làm baseline nghiên cứu, không đáp ứng yêu cầu độ chính xác cao của hệ thống triển khai thực tế.

Các phương pháp hiện đại dựa trên hàm mất mát biên góc như ArcFace, CosFace và SphereFace được xem là hướng tiếp cận tiên tiến (state-of-the-art) trong nhận diện khuôn mặt hiện nay. Những mô hình này tối ưu hóa khoảng cách góc giữa các danh tính, giúp embedding phân tách rõ ràng hơn trong không gian đặc trưng và thường được chuẩn hóa trên hypersphere, rất phù hợp cho việc sử dụng độ tương đồng cosine. Trong số đó, ArcFace là mô hình được công bố năm 2019 và đến nay vẫn được xem là chuẩn tham chiếu trong nhiều hệ thống thương mại và nghiên cứu. ArcFace tạo ra embedding 512 chiều với độ phân biệt cao, ổn định trước sự thay đổi về ánh sáng, góc quay, biểu cảm và độ tuổi.

Ưu điểm nổi bật của ArcFace là độ chính xác cao trong các bộ dữ liệu in-the-wild, khả năng xử lý tốt bài toán one-to-many ở quy mô lớn và khả năng tích hợp hiệu quả với các công cụ tìm kiếm vector như FAISS cũng như các nền tảng tăng tốc suy luận như ONNX Runtime và GPU. Mặc dù mô hình có kích thước lớn hơn và embedding 512 chiều tiêu tốn bộ nhớ nhiều hơn so với các mô hình 128 chiều, nhưng chi phí này được đánh đổi bằng độ tin cậy và khả năng mở rộng vượt trội. Do đó, ArcFace được lựa chọn làm phương án trích xuất đặc trưng chính thức cho hệ thống Face ID trong luận văn, đáp ứng tốt yêu cầu về độ chính xác, tính ổn định và khả năng triển khai trong môi trường bệnh viện thực tế.

3.2.2.3. Khối Matching – Đối sánh và ra quyết định

Khối Matching so sánh embedding của ảnh truy vấn với cơ sở dữ liệu embedding và đưa ra đánh giá cuối cùng. Để tối ưu tốc độ và độ chính xác, hệ thống sử dụng cơ chế lọc hai lớp, trong đó:

- Lớp 1: tìm kiếm thô bằng FAISS để lấy top-K ứng viên;
- Lớp 2: xác thực tinh bằng cosine similarity để đưa ra kết luận.

Cấu trúc này phù hợp cho hệ thống bệnh viện với số lượng bệnh nhân lớn và yêu cầu phản hồi nhanh.

3.2.3. Quy trình xây dựng và lựa chọn mô hình Face ID tối ưu

Quy trình xây dựng và lựa chọn mô hình Face ID được thực hiện theo hướng thực nghiệm có kiểm soát, nhằm đảm bảo hệ thống đạt được đồng thời ba mục tiêu quan trọng: độ chính xác cao, tốc độ xử lý phù hợp thời gian thực và khả năng mở rộng trong môi trường bệnh viện. Toàn bộ pipeline nhận diện được chia thành hai lớp chính là Coarse Search và Fine Verification. Trong đó, lớp Coarse Search đóng vai trò lọc nhanh các ứng viên tiềm năng từ cơ sở dữ liệu lớn, còn lớp Fine Verification đảm nhiệm việc xác thực chính xác danh tính bệnh nhân ở mức 1:1.

Việc lựa chọn mô hình và cấu hình tối ưu cho từng lớp không được thực hiện dựa trên lý thuyết thuần túy, mà thông qua quá trình thử nghiệm, đo lường và so sánh nhiều phương án khác nhau trong điều kiện dữ liệu và phần cứng thực tế của hệ thống.

3.2.3.1. Phân tích các phương án Coarse Search lọc lấy top-k

a. Thiết kế và mô tả các phương án coarse search

Mục tiêu của lớp Coarse Search là lọc ra một tập ứng viên top-K từ cơ sở dữ liệu embedding sao cho thỏa mãn hai yêu cầu bắt buộc. Thứ nhất, đảm bảo 100% trường hợp khuôn mặt đúng (ground truth) phải xuất hiện trong danh sách top-K, nếu khuôn mặt đầu vào thực sự tồn tại trong cơ sở dữ liệu. Thứ hai, thời gian tìm kiếm phải đủ nhanh để hệ thống có thể xử lý liên tục nhiều truy vấn trong giờ cao điểm của bệnh viện.

Để đạt được mục tiêu này, cần triển khai và so sánh nhiều kỹ thuật tìm kiếm khác nhau, bao gồm:

- Brute-force L2
- Brute-force cosine similarity
- FAISS IndexFlatL2
- FAISS IndexIVFFlat
- Các biến thể kết hợp PCA và FAISS

Việc thử nghiệm nhiều phương án cho phép đánh giá toàn diện về tốc độ, độ bao phủ (recall) và khả năng mở rộng trước khi lựa chọn cấu hình vận hành chính thức.

b. Phân tích từng phương án coarse search

Brute-force L2 là phương pháp cơ bản nhất, trong đó khoảng cách Euclid (L2) được tính trực tiếp giữa embedding truy vấn và toàn bộ embedding trong cơ sở dữ liệu. Phương án này được sử dụng như một baseline chuẩn để so sánh, do cho kết quả chính xác tuyệt đối và không phụ thuộc vào chỉ mục tìm kiếm.

Ưu điểm của brute-force L2 là độ chính xác 100%, kết quả ổn định và dễ triển khai. Tuy nhiên, nhược điểm lớn nhất là tốc độ xử lý chậm do độ phức tạp tuyến tính O(N). Khi số lượng bệnh nhân tăng lên hàng chục nghìn, phương pháp này không còn đáp ứng yêu cầu thời gian thực của hệ thống bệnh viện.

Brute-force cosine similarity được đưa vào thử nghiệm do cosine similarity là độ đo phổ biến trong các hệ thống nhận diện khuôn mặt, đặc biệt với embedding đã chuẩn hóa như ArcFace. So với L2, phương pháp này có tốc độ nhỉnh hơn đôi chút và phù hợp hơn với embedding có hướng. Tuy nhiên, do vẫn mang bản chất brute-force, tốc độ xử lý vẫn giảm tuyến tính theo kích thước cơ sở dữ liệu và không phù hợp để vận hành lâu dài ở quy mô lớn.

FAISS IndexFlatL2 là phương pháp tìm kiếm chính xác (exact search) của thư viện FAISS, cho kết quả tương đương brute-force nhưng được tối ưu hóa bằng SIMD, đa luồng và quản lý bộ nhớ hiệu quả. Phương án này được đưa vào thử nghiệm nhằm đánh giá mức độ cải thiện tốc độ so với brute-force trong khi vẫn giữ nguyên độ chính xác.

Kết quả cho thấy IndexFlatL2 nhanh hơn brute-force khoảng một bậc độ lớn (khoảng 10 lần), đạt độ chính xác 100% và không yêu cầu huấn luyện chỉ mục. Phương pháp này ổn định khi số lượng embedding tăng từ 10.000 lên 50.000 hoặc hơn, đồng thời dễ tích hợp vào pipeline hiện tại.

FAISS IndexIVFFlat là phương pháp tìm kiếm xấp xỉ (Approximate Nearest Neighbor), sử dụng cơ chế phân cụm để thu hẹp không gian tìm kiếm. IVFFlat cho tốc độ truy vấn rất cao, tuy nhiên đánh đổi bằng việc giảm độ bao phủ (recall). Trong thử nghiệm, phương án này đạt tốc độ nhanh nhất nhưng không đảm bảo 100% Recall@10, dẫn đến nguy cơ bỏ sót ứng viên đúng, điều không thể chấp nhận trong hệ thống y tế.

PCA kết hợp FAISS được thử nghiệm nhằm giảm chiều embedding từ 128D xuống 64D hoặc 32D, với mục tiêu tăng tốc tìm kiếm và giảm dung lượng bộ nhớ. Mặc dù PCA giúp giảm kích thước dữ liệu, kết quả thực nghiệm cho thấy lợi ích về tốc độ không đáng kể, trong khi biên phân biệt của embedding bị suy giảm, làm giảm độ an toàn của lớp lọc thô.

c. Kết quả thực nghiệm và phân tích

Các phương án coarse search được đánh giá trên tập 29 ảnh truy vấn đại diện cho nhiều tình huống khác nhau, với cơ sở dữ liệu gồm 10.473 embedding bệnh nhân. Các tiêu chí đánh giá bao gồm Recall@10, thời gian truy vấn trung bình, cosine similarity top-1 và trade-off giữa tốc độ và độ bao phủ.

Kết quả cho thấy các phương án exact search (Brute-force L2, Brute-force cosine và FAISS IndexFlatL2) đều đạt 100% Recall@10, trong khi IVFFlat chỉ đạt 96.6% do bản chất tìm kiếm xấp xỉ. Về tốc độ, IVFFlat nhanh nhất, tiếp theo là IndexFlatL2, còn

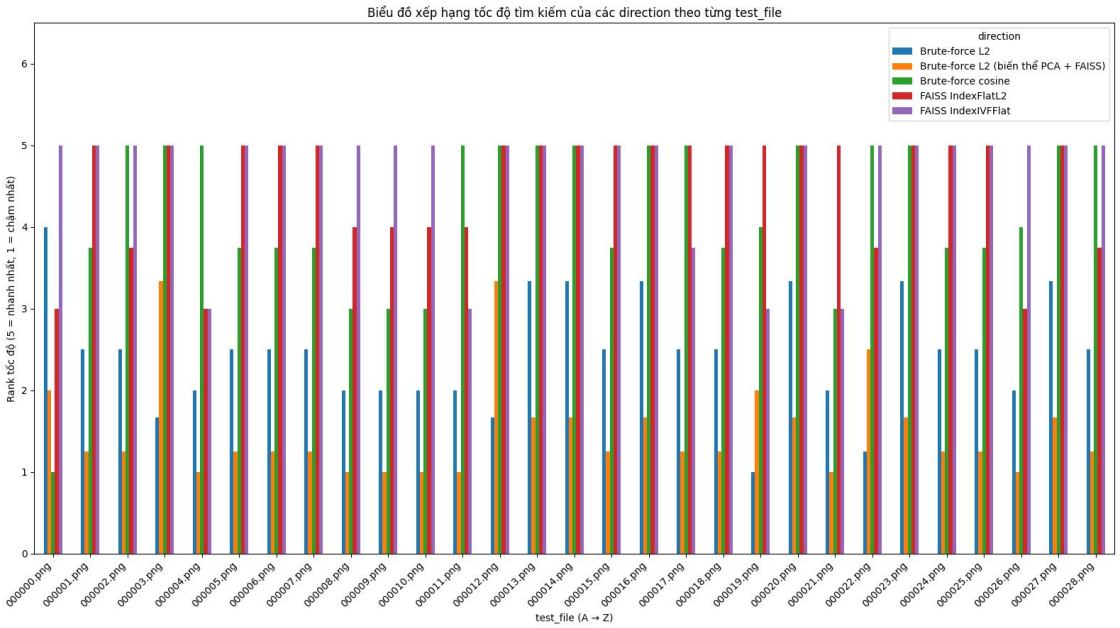
các phương án brute-force chậm hơn đáng kể. PCA không mang lại lợi ích rõ ràng về tốc độ và làm suy giảm đặc trưng embedding.

Bảng 3.1. Kết quả thực nghiệm với 5 phương án coarse search

Phương án	Recall@10 (%)	Tốc độ trung bình (s)	Cosine top-1 trung bình	Ghi chú
Brute-force L2	100	0.0050	0.968	Chính xác tuyệt đối nhưng chậm; baseline chuẩn.
Brute-force Cosine	100	0.0044	0.968	Nhanh hơn L2 nhờ chuẩn hóa vector; vẫn không đạt yêu cầu real-time khi DB lớn.
FAISS IndexFlatL2	100	0.00053	0.831	Tăng tốc ~10 lần; exact search; ổn định, dễ triển khai.
FAISS IndexIVFFlat	96.6	0.00024	0.831	Nhanh nhất (~20× brute), nhưng giảm recall do tìm kiếm xấp xỉ.
PCA + FAISS (FlatL2)	100	0.0057	0.968	Recall tốt nhưng tốc độ không cải thiện; PCA gây mất thông tin nhẹ..

Các phương án coarse search được đánh giá trên tập 29 ảnh truy vấn đại diện cho nhiều tình huống khác nhau, với cơ sở dữ liệu gồm 10.473 embedding bệnh nhân. Các tiêu chí đánh giá bao gồm Recall@10, thời gian truy vấn trung bình, cosine similarity top-1 và trade-off giữa tốc độ và độ bao phủ.

Kết quả cho thấy các phương án exact search (Brute-force L2, Brute-force cosine và FAISS IndexFlatL2) đều đạt 100% Recall@10, trong khi IVFFlat chỉ đạt 96.6% do bản chất tìm kiếm xấp xỉ. Về tốc độ, IVFFlat nhanh nhất, tiếp theo là IndexFlatL2, còn các phương án brute-force chậm hơn đáng kể. PCA không mang lại lợi ích rõ ràng về tốc độ và làm suy giảm đặc trưng embedding.



Hình 3.1 Biểu đồ mô tả tốc độ tìm kiếm của 5 phương án coarse search

Biểu đồ ở Hình 3.1 minh họa rõ thứ hạng tốc độ của các phương án coarse search trên toàn bộ tập truy vấn. FAISS IndexIVFFlat luôn đạt tốc độ cao nhất, IndexFlatL2 ổn định ở nhóm nhanh thứ hai, trong khi brute-force và PCA nằm ở nhóm chậm nhất.

d. Kết luận lựa chọn phương án coarse search

Từ các phân tích và kết quả thực nghiệm, lựa chọn FAISS IndexFlatL2 với embedding 128 chiều, không áp dụng PCA làm phương án coarse search chính thức cho hệ thống. Lựa chọn này đáp ứng đầy đủ các yêu cầu đặt ra, bao gồm 100% Recall@10, tốc độ truy vấn nhanh hơn brute-force khoảng 10 lần (0.00053 giây mỗi truy vấn), độ tách biệt đặc trưng đủ tốt để phục vụ lớp Fine Verification, và khả năng mở rộng linh hoạt khi thêm bệnh nhân mới mà không cần huấn luyện lại chỉ mục.

Phương án IVFFlat không được lựa chọn do chưa đảm bảo độ an toàn tuyệt đối về recall, trong khi PCA không mang lại lợi ích rõ ràng trong quy mô dữ liệu hiện tại và có thể làm suy giảm độ tin cậy của hệ thống.

3.2.3.2. Phân tích các phương án Fine Verification

a. Thiết kế và mô tả các phương án fine verification

Sau khi lớp lọc thô (Level 1 – Coarse Search) truy xuất được danh sách top-K ứng viên có khả năng trùng khớp cao, hệ thống tiếp tục thực hiện lớp lọc tinh (Level 2 – Fine Verification). Nhiệm vụ của lớp này là xác thực theo mô hình 1:1 giữa ảnh truy vấn và từng embedding ứng viên nhằm đưa ra kết luận cuối cùng về việc khuôn mặt đầu vào có tồn tại trong cơ sở dữ liệu bệnh viện hay không, và nếu có thì xác định chính xác danh tính bệnh nhân tương ứng.

Do yêu cầu của hệ thống y tế là độ tin cậy cao và tuyệt đối không được trả nhầm danh tính, các phương pháp Fine Verification cần được lựa chọn và đánh giá một cách cẩn trọng. Trong phạm vi luận văn, hai hướng tiếp cận đại diện cho hai thế hệ kỹ thuật embedding khác nhau được đưa vào thực nghiệm, bao gồm dlib_resnet và arcface_insightface. Việc so sánh hai phương án này giúp luận văn có cơ sở khoa học rõ ràng trước khi lựa chọn mô hình vận hành chính thức cho hệ thống.

1. Phương án 1: dlib_resnet (128D)

Mô hình Dlib ResNet là kiến trúc embedding phổ biến trong thư viện face_recognition, tạo ra vector đặc trưng có kích thước 128 chiều. Mô hình này được huấn luyện trên các bộ dữ liệu nhận diện khuôn mặt cổ điển như LFW và các biến thể liên quan, do đó có mức độ ổn định nhất định trong các điều kiện chụp tiêu chuẩn.

Trong lớp Fine Verification, phương án này sử dụng khoảng cách Euclid (L2 distance) để so sánh embedding truy vấn với embedding của từng ứng viên trong danh sách top-K. Kết quả so khớp được quyết định dựa trên việc so sánh với một ngưỡng định sẵn, thường nằm trong khoảng từ 0.6 đến 0.8.

Ưu điểm của phương án dlib_resnet là mô hình nhẹ, không yêu cầu GPU và có thể chạy ổn định trên hầu hết các máy chủ phổ thông. Tốc độ xử lý nhanh khi số lượng ứng viên top-K nhỏ (khoảng 10–100), dễ tích hợp và ít phụ thuộc vào các thư viện bên ngoài. Mô hình này hoạt động tương đối ổn định trong điều kiện khuôn mặt chính diện và ánh sáng tốt.

Tuy nhiên, hạn chế của dlib_resnet là khả năng phân biệt danh tính (discriminative power) thấp hơn so với các mô hình hiện đại như ArcFace. Hiệu quả nhận diện giảm rõ rệt trong các trường hợp khuôn mặt nghiêng, biểu cảm thay đổi hoặc điều kiện ánh sáng yếu và không đồng đều. Do biên phân tách giữa các danh tính nhỏ, mô hình dễ phát sinh lỗi false positive khi cơ sở dữ liệu lớn (trên khoảng 5.000 người). Ngoài ra, embedding chỉ có 128 chiều nên lượng thông tin đặc trưng bị hạn chế, không tối ưu cho bài toán xác thực tinh trong môi trường bệnh viện.

2. Phương án 2: arcface_insightface (512D)

ArcFace là mô hình nhận diện khuôn mặt hiện đại, sử dụng hàm mất mát Additive Angular Margin Loss nhằm tối ưu hóa khoảng cách góc giữa các danh tính trong không gian embedding. Mô hình này tạo ra vector đặc trưng có kích thước 512 chiều, cho phép biểu diễn khuôn mặt một cách phong phú và ổn định trong nhiều điều kiện thực tế khác nhau. Các mô hình được huấn luyện sẵn của InsightFace, tiêu biểu là buffalo_l, được sử dụng cho quá trình Fine Verification.

Trong bước xác thực tinh, ArcFace sử dụng độ đo cosine similarity để đánh giá mức độ tương đồng giữa embedding truy vấn và embedding ứng viên, sau đó so sánh với ngưỡng quyết định (thường nằm trong khoảng 0.4–0.65, tùy theo yêu cầu an toàn

của hệ thống). Cách tiếp cận này đặc biệt phù hợp với bài toán one-to-many trong các hệ thống nhận diện quy mô lớn.

Ưu điểm nổi bật của ArcFace là độ phân biệt danh tính rất cao, được đánh giá là một trong những mô hình tốt nhất hiện nay cho nhận diện khuôn mặt. Mô hình cho kết quả ổn định trong nhiều điều kiện thực tế như khuôn mặt nghiêng, thay đổi ánh sáng, khác biệt biểu cảm và chất lượng ảnh không đồng đều. Biên phân tách giữa các danh tính rõ rệt giúp giảm thiểu tối đa khả năng false positive. Ngoài ra, ArcFace tương thích tốt với cosine similarity và thư viện FAISS, thuận lợi cho việc tích hợp trong hệ thống tìm kiếm và xác thực nhiều tầng.

Bên cạnh đó, hạn chế của phương án này là kích thước embedding lớn hơn (512 chiều), dẫn đến nhu cầu bộ nhớ cao hơn so với dlib_resnet. Tốc độ xử lý trên CPU chậm hơn đôi chút, tuy nhiên do số lượng ứng viên top-K trong lớp Fine Verification tương đối nhỏ nên vẫn đáp ứng tốt yêu cầu thời gian thực của hệ thống. Hiệu năng của mô hình được tối ưu nhất khi chạy trên GPU, mặc dù trong bối cảnh hệ thống hiện tại, việc chạy trên CPU vẫn đạt hiệu quả chấp nhận được.

b. Kết quả thực nghiệm và lựa chọn phương án fine verification tối ưu

Thực nghiệm được tiến hành trên tập dữ liệu gồm 29 ảnh truy vấn, ký hiệu từ file_000000.png đến file_000028.png. Với mỗi ảnh truy vấn, hệ thống thực hiện bước Coarse Search để lựa chọn top-5 ứng viên có độ tương đồng cao nhất từ cơ sở dữ liệu, sau đó tiến hành so sánh chi tiết theo mô hình nhận diện tương ứng. Kết quả thực nghiệm thu được 29 cặp so khớp đúng (self-match), trong đó ảnh truy vấn được so khớp với chính đối tượng tương ứng trong cơ sở dữ liệu, và 116 cặp so khớp sai (non-match) giữa ảnh truy vấn và các ứng viên không cùng danh tính. Như vậy, tổng cộng có 145 cặp so sánh 1:1 được sử dụng để đánh giá cho mỗi mô hình nhận diện.

Các metrics được sử dụng:

- Accuracy (%) – tỷ lệ quyết định đúng.
- Cosine của match và non-match – đánh giá độ phân biệt.
- Margin = Cosine(match) − Cosine(non-match).
- FPR@threshold – tỷ lệ non-match vượt qua ngưỡng cosine.
- Similarity percent – dùng cho ArcFace.

Bảng 3.2. Kết quả thực nghiệm với 2 phương án fine verification

Phương án	Accuracy (%)	Cosine match (mean)	Cosine non-match (mean)	Margin	L2 match	L2 non	FPR@threshold

Dlib ResNet (128D)	96.55	0.967	0.901	0.066	0.256	0.444	0% (thr=0.6)
ArcFace InsightFace (512D)	100	0.681	0.110	0.571	0.795	1.334	0% (thr=0.4)

Phân tích kết quả

1. Accuracy

Kết quả thực nghiệm cho thấy mô hình ArcFace đạt độ chính xác 100%, tức là toàn bộ các trường hợp self-match đều được nhận diện đúng và không xuất hiện bất kỳ trường hợp nhận diện nhầm (false positive) nào trong tập dữ liệu thử nghiệm. Điều này cho thấy khả năng phân biệt rõ ràng giữa các đối tượng cùng danh tính và khác danh tính của mô hình.

Ngược lại, mô hình Dlib chỉ đạt độ chính xác 96,55%, trong đó xuất hiện ít nhất một trường hợp self-match nhưng giá trị cosine similarity không vượt qua ngưỡng phân loại, dẫn đến việc hệ thống không xác nhận đúng danh tính. Trong bối cảnh ứng dụng y tế, nơi yêu cầu độ chính xác gần như tuyệt đối và không cho phép sai lệch danh tính bệnh nhân, sự khác biệt này mang ý nghĩa quyết định đối với khả năng triển khai thực tế của hệ thống.

2. Separation Margin (Biên phân tách)

Biên phân tách giữa các trường hợp match và non-match của mô hình ArcFace đạt giá trị 0,571, lớn gấp khoảng 8,6 lần so với biên phân tách của mô hình Dlib (0,066). Giá trị margin lớn cho thấy hai phân bố similarity của ArcFace được tách biệt rõ ràng, từ đó giúp việc lựa chọn ngưỡng phân loại (threshold) trở nên đơn giản và an toàn hơn trong vận hành.

Ngược lại, margin nhỏ của Dlib phản ánh mức độ chồng lấn lớn giữa các trường hợp match và non-match, khiến hệ thống nhạy cảm với sai số và khó thiết lập ngưỡng phân loại ổn định. Kết quả này cho thấy ArcFace vượt trội hơn Dlib trong việc giảm thiểu rủi ro nhận diện nhầm.

3. Cosine Non-match

Đối với mô hình ArcFace, giá trị cosine similarity trung bình của các trường hợp non-match xấp xỉ 0,110, nằm ở mức rất thấp. Điều này cho thấy các đối tượng khác nhau trong cơ sở dữ liệu có độ tương đồng nhỏ, nguy cơ nhầm lẫn giữa các cá thể là không đáng kể, ngay cả khi số lượng đối tượng trong hệ thống tăng lên.

Trong khi đó, mô hình Dlib cho giá trị cosine similarity trung bình của các trường hợp non-match vào khoảng 0,901, ở mức rất cao. Điều này đồng nghĩa với việc nhiều trường hợp impostor có độ tương đồng tiệm cận với các trường hợp so khớp đúng, làm

gia tăng đáng kể nguy cơ false positive khi cơ sở dữ liệu mở rộng. Đây là một hạn chế nghiêm trọng đối với các hệ thống yêu cầu độ an toàn định danh cao.

4. L2 khoảng cách

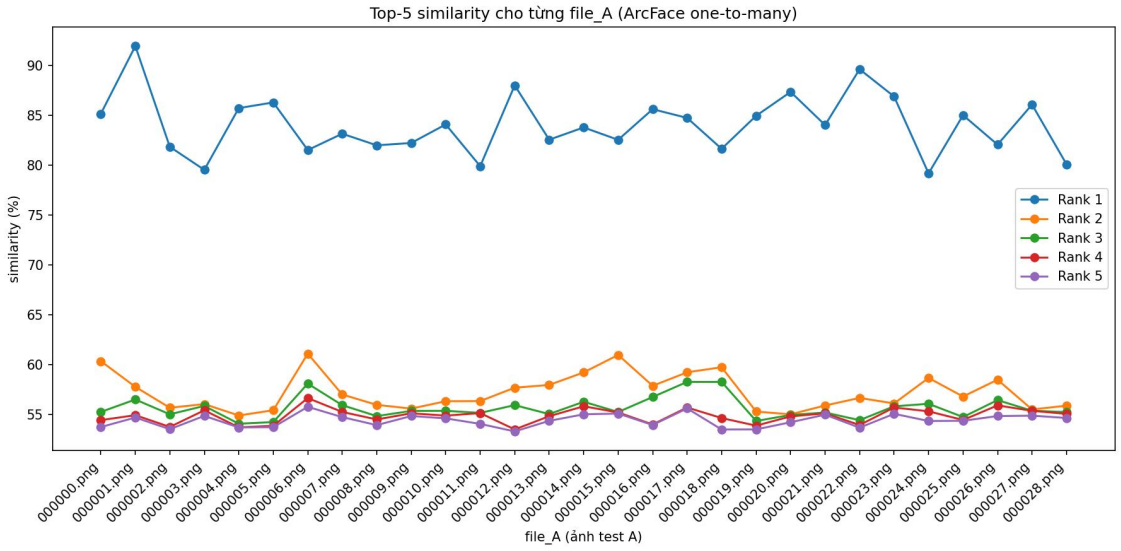
Phân tích khoảng cách L2 cho thấy mô hình ArcFace tạo ra sự khác biệt rõ ràng giữa các trường hợp match và non-match, với giá trị trung bình lần lượt là 0,795 và 1,334. Khoảng cách chênh lệch lớn này giúp việc phân loại trở nên trực quan và ít phụ thuộc vào tinh chỉnh ngưỡng.

Ngược lại, đối với mô hình Dlib, khoảng cách L2 giữa match (0,256) và non-match (0,444) khá nhỏ, dẫn đến nguy cơ chồng lấn và sai lệch khi lựa chọn ngưỡng phân loại. Điều này tiếp tục khẳng định hạn chế của Dlib trong việc triển khai ở các hệ thống nhận diện yêu cầu độ tin cậy cao.

Kết quả thực nghiệm cho thấy:

- ArcFace sở hữu embedding có tính phân biệt mạnh, ổn định.
- Dlib có xu hướng chồng lấn match/non-match → không phù hợp triển khai thực tế.

Tuy nhiên, để hiểu sâu hơn cơ chế phân tách của hai mô hình, cần xem trực quan từ hai biểu đồ similarity.



Hình 3.2 Biểu đồ mô tả kết quả thực nghiệm của ArcFace

Nhận xét và đánh giá kết quả ArcFace

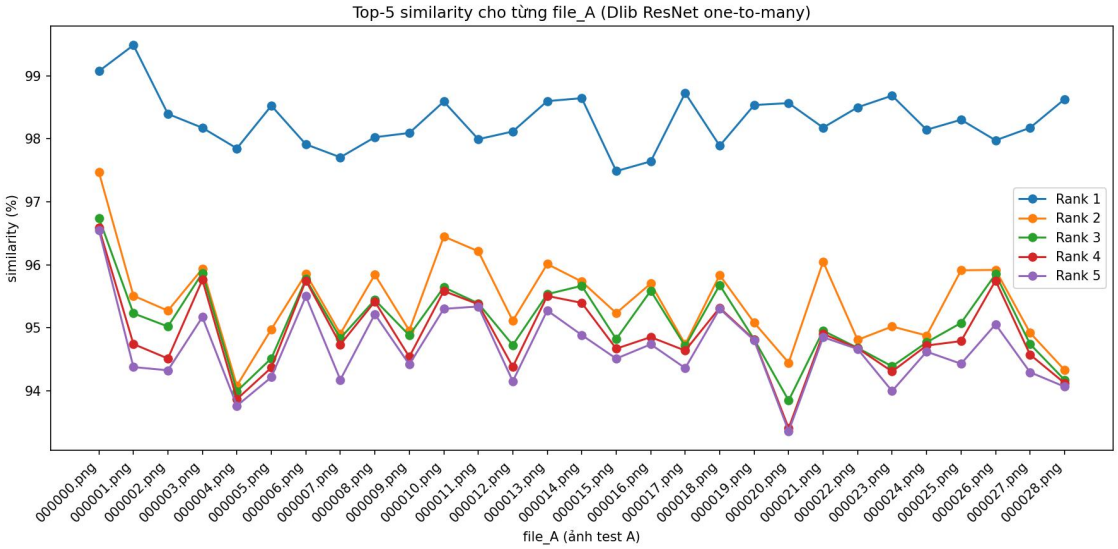
Kết quả thực nghiệm cho thấy các mẫu có thứ hạng 1 (ứng viên đứng đầu trong danh sách trả về) luôn đạt giá trị similarity cao và ổn định, dao động trong khoảng 80–92%. Điều này thể hiện rõ khả năng self-match chính xác và nhất quán của mô hình

ArcFace, khi ảnh truy vấn được so khớp đúng với đối tượng tương ứng trong cơ sở dữ liệu.

Ngược lại, các mẫu ở thứ hạng từ 2 đến 5 tập trung trong một dải giá trị similarity thấp hơn, khoảng 53–60%. Các kết quả này đại diện cho các trường hợp non-match (impostor) và nằm cách xa cụm self-match, tạo ra một margin phân tách rõ ràng giữa hai nhóm. Sự tách biệt này cho thấy mô hình có khả năng phân biệt tốt giữa các khuôn mặt cùng danh tính và khác danh tính, ngay cả khi số lượng ứng viên được mở rộng trong bước Coarse Search.

Đáng chú ý, trong toàn bộ tập thực nghiệm không có trường hợp non-match nào vượt qua ngưỡng 70% similarity. Điều này giúp việc thiết lập ngưỡng vận hành (threshold) trong hệ thống trở nên đơn giản và ổn định, đồng thời giảm nguy cơ nhiễu hoặc chồng lấn giữa các lớp.

Tổng hợp các kết quả trên cho thấy phân bố similarity của mô hình ArcFace hình thành hai cụm tách biệt hoàn toàn: cụm match nằm trong khoảng 80–92% và cụm non-match nằm trong khoảng 53–60%. Đây là một đặc tính đặc biệt quan trọng trong bối cảnh ứng dụng y tế, nơi hệ thống nhận diện không được phép trả nhầm danh tính bệnh nhân, bởi sai sót trong định danh có thể dẫn đến hậu quả nghiêm trọng trong quá trình khám và điều trị.



Hình 3.3 Biểu đồ mô tả kết quả thực nghiệm của Dlib

Nhận xét và đánh giá kết quả đối với mô hình Dlib

Kết quả thực nghiệm cho thấy các mẫu ở thứ hạng 1 của mô hình Dlib đạt giá trị similarity rất cao, dao động trong khoảng 98–99,5%. Tuy nhiên, đáng chú ý là các mẫu ở thứ hạng từ 2 đến 5 cũng có giá trị similarity ở mức cao, khoảng 94–96%. Điều này cho thấy các trường hợp non-match (impostor) nằm rất gần vùng của match, dẫn đến hiện tượng chồng lấn phân bố similarity ở mức độ lớn.

Phân tích chi tiết hơn cho thấy tồn tại các trường hợp non-match có giá trị similarity gần tương đương với match. Cụ thể, có những cặp non-match đạt giá trị similarity lên tới 0,949, trong khi một số cặp match chỉ đạt khoảng 0,960. Sự chênh lệch nhỏ này khiến việc lựa chọn ngưỡng phân loại (threshold) trở nên đặc biệt khó khăn. Nếu ngưỡng không được thiết lập ở mức cực kỳ chặt chẽ, hệ thống sẽ rất dễ phát sinh false positive, tức là nhận diện nhầm danh tính.

Từ các kết quả trên có thể thấy rằng, mặc dù mô hình Dlib cho ra giá trị similarity cao, nhưng khả năng phân tách giữa match và non-match không rõ ràng. Đặc tính này tiềm ẩn rủi ro nghiêm trọng khi triển khai trong môi trường thực tế, đặc biệt là trong các ứng dụng y tế, nơi yêu cầu về độ chính xác và an toàn định danh là rất cao.

c. Kết luận lựa chọn phương án Fine Verification

Kết quả đánh giá định lượng và phân tích biểu đồ similarity cho thấy sự khác biệt rõ rệt giữa hai mô hình embedding được thử nghiệm. ArcFace InsightFace (512D) thể hiện ưu thế vượt trội ở cả độ chính xác, biên phân tách và độ ổn định, trong khi Dlib ResNet (128D) bộc lộ hạn chế đáng kể khi cơ sở dữ liệu mở rộng.

Thứ nhất, ArcFace đạt độ chính xác tuyệt đối (100%), không xảy ra bất kỳ trường hợp nhận diện sai nào trong toàn bộ 145 phép so sánh, trong khi Dlib chỉ đạt 96,55% do xuất hiện self-match bị loại bởi threshold. Ở môi trường bệnh viện, nơi yêu cầu quan trọng nhất là không được phép nhầm danh tính bệnh nhân, sự khác biệt nhỏ về độ chính xác cũng mang ý nghĩa quyết định.

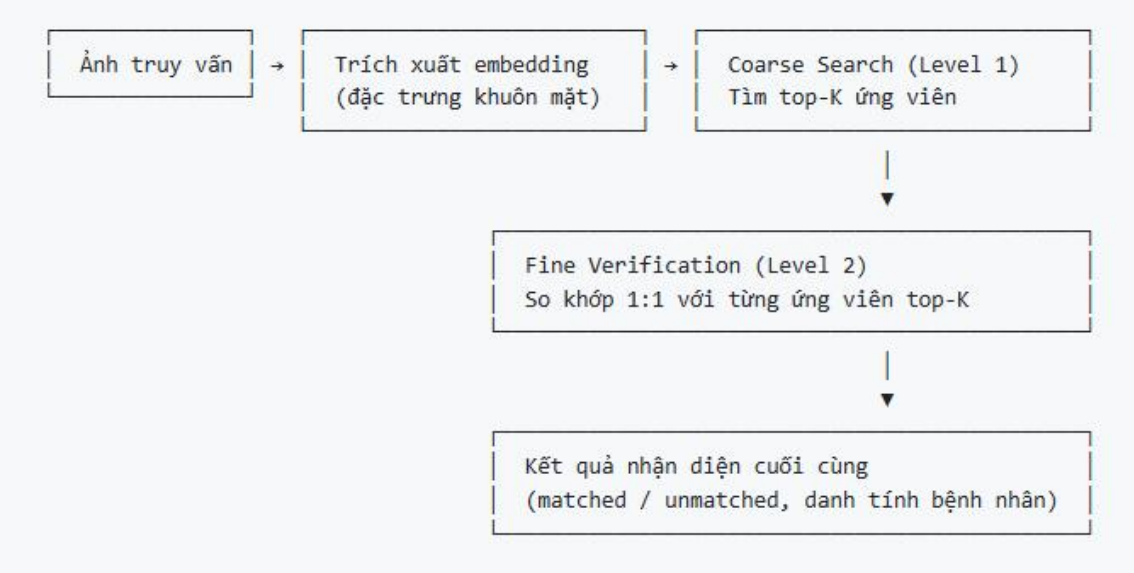
Thứ hai, biên phân tách (separation margin) giữa match và non-match của ArcFace vượt trội (0,571 so với 0,066 của Dlib). Các biểu đồ similarity càng củng cố nhận định này: ArcFace tạo nên hai cụm giá trị tách biệt hoàn toàn—self-match nằm trong vùng 80–92%, còn non-match giữ ổn định trong khoảng 53–60%. Ngược lại, Dlib cho thấy mức similarity rất cao ở cả match và non-match (94–99%), dẫn đến hiện tượng chồng lấn nghiêm trọng. Điều này khiến Dlib tiềm ẩn rủi ro false positive rất lớn khi cơ sở dữ liệu tăng lên hàng chục nghìn bệnh nhân.

Thứ ba, ArcFace duy trì độ ổn định cao trong nhiều điều kiện chụp khác nhau (pose, ánh sáng, biểu cảm), đồng thời tương thích tối ưu với cơ chế đối sánh cosine và pipeline lọc hai lớp. Mặc dù embedding 512 chiều khiến thời gian xử lý lớn hơn so với Dlib, mức tăng này là rất nhỏ và hoàn toàn chấp nhận được trong bối cảnh số lượng ứng viên sau coarse search chỉ còn khoảng top-5 đến top-10.

Từ những phân tích trên, có thể khẳng định rằng ArcFace InsightFace (512D) là lựa chọn tối ưu cho lớp Fine Verification, bảo đảm độ tin cậy tuyệt đối, biên an toàn lớn và khả năng vận hành ổn định trong hệ thống thực tế. Ngưỡng vận hành phù hợp được đề xuất là cosine > 0,5 (tương đương ~70% similarity), vừa đủ chặt để tránh nhầm lẫn, vừa không loại bỏ self-match trong điều kiện thực tế.

Ngược lại, Dlib ResNet—dù có tốc độ tốt—chỉ phù hợp cho giai đoạn thử nghiệm hoặc làm baseline so sánh, không đạt yêu cầu triển khai trong môi trường bệnh viện nơi độ chính xác và an toàn phải được đặt lên hàng đầu.

3.2.4. Thiết kế cơ chế lọc hai lớp(coarse search + fine verification)



Hình 3.4 Minh họa cơ chế lọc coarse search + fine verification

Hình 3.4. Sơ đồ minh họa cơ chế "lọc hai lớp" trong nhận diện khuôn mặt, trong đó bước Coarse Search thu hẹp tập ứng viên top-K và bước Fine Verification thực hiện xác thực chi tiết từng ứng viên để đưa ra quyết định cuối cùng.

Dựa trên kiến trúc ba khối Detection – Embedding – Matching (mục 3.2.2) và kết quả đánh giá chi tiết các phương án coarse search và fine verification (mục 3.2.3), mô-đun Face ID của hệ thống được hiện thực hóa theo cơ chế lọc hai lớp:

- Lớp lọc 1 (Level 1 – Coarse Search): tìm kiếm nhanh trên toàn bộ cơ sở dữ liệu embedding để thu về tập ứng viên hẹp top-K nhưng vẫn đảm bảo không bỏ sót trường hợp đúng.
- Lớp lọc 2 (Level 2 – Fine Verification): xác thực tinh 1:1 giữa embedding truy vấn và từng ứng viên top-K, từ đó đưa ra quyết định cuối cùng "có/không tồn tại" và danh tính bệnh nhân.

Cách tiếp cận này cho phép tách rời hai mục tiêu thường mâu thuẫn trong hệ thống nhận diện khuôn mặt:

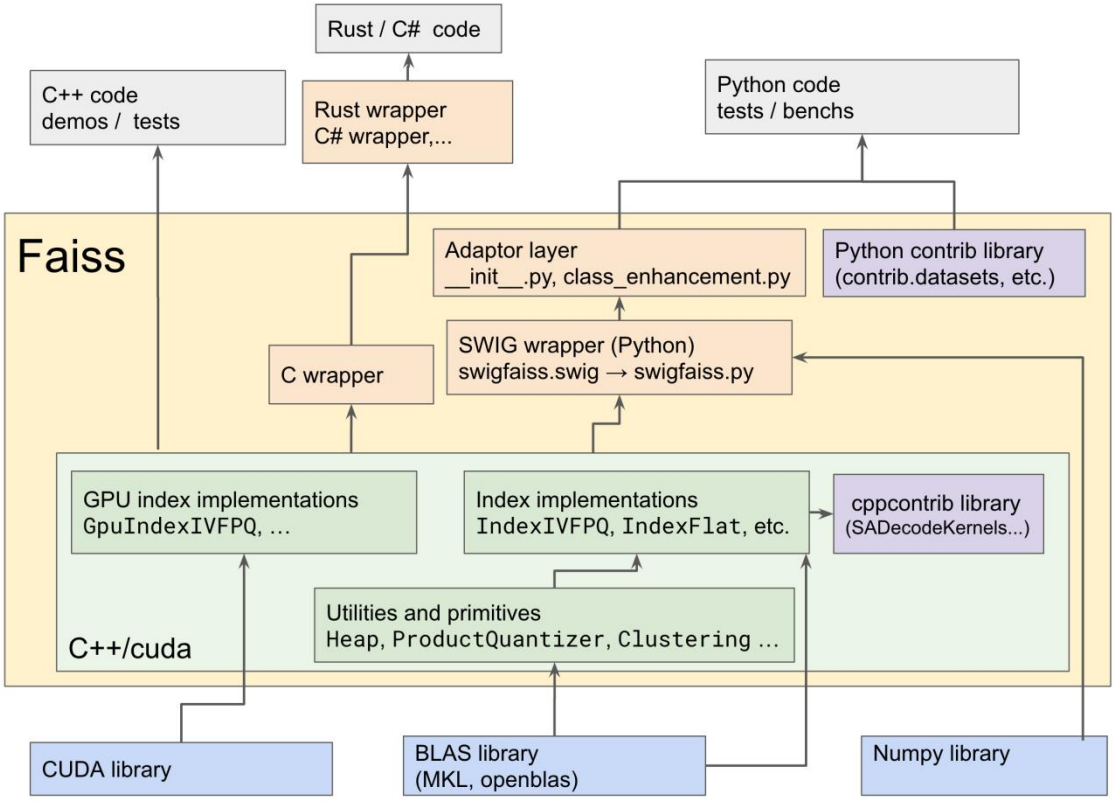
- Tốc độ khi cơ sở dữ liệu lớn.

- Độ tin cậy ở bước ra quyết định cuối cùng. Lớp 1 ưu tiên tối đa tốc độ với yêu cầu recall cao, trong khi lớp 2 ưu tiên biên phân tách và độ chính xác tuyệt đối.

3.2.4.1. Lớp lọc 1 – Coarse Search bằng FAISS IndexFlatL2

Từ kết quả thực nghiệm ở mục 3.2.3.1, FAISS IndexFlatL2 được lựa chọn làm kỹ thuật coarse search chính thức nhờ đạt đồng thời:

- Recall@10 = 100%: top-10 luôn chứa đúng ground truth, đáp ứng yêu cầu "không bỏ sót" ứng viên thật.
- Thời gian truy vấn trung bình ~0,00053 s/truy vấn: nhanh hơn khoảng một bậc độ lớn so với brute-force, phù hợp cho môi trường vận hành thời gian thực.
- Cấu hình đơn giản, không cần huấn luyện index, dễ bảo trì và mở rộng.



Hình 3.5 Minh họa cơ chế lọc coarse search bằng FAISS IndexFlatL2

Hình 3.5 minh họa cơ chế lọc coarse search sử dụng FAISS IndexFlatL2 trong mô-đun Face ID. Trong triển khai cuối cùng, hệ thống sử dụng vector embedding ArcFace 512 chiều cho cả cơ sở dữ liệu và ảnh truy vấn; tuy nhiên, chiến lược tìm kiếm thô vẫn được giữ nguyên, dựa trên việc tìm kiếm các lân cận gần nhất trong không gian embedding bằng IndexFlatL2. Việc thay đổi mô hình sinh embedding từ

Dlib sang ArcFace chỉ làm thay đổi phân bố của các vector đặc trưng, trong khi bản chất của thuật toán tìm kiếm và cơ chế coarse search không bị ảnh hưởng.

Quy trình xử lý ở lớp lọc 1 như sau:

1. Ảnh đầu vào được khối Detection (YOLO-face) cắt vùng mặt, sau đó khối Embedding sinh ra vector chuẩn hóa bằng mô hình ArcFace (InsightFace).
2. FAISS IndexFlatL2 nhận vector chuẩn hóa và thực hiện phép tìm kiếm lân cận gần nhất trên toàn bộ tập embedding đã được build trước đó, trả về hai mảng:
   + D: khoảng cách L2 tới các vector gần nhất;
   + I: chỉ số các vector tương ứng trong cơ sở dữ liệu.
3. Hệ thống lấy top-K ứng viên làm đầu vào cho lớp lọc 2.

Với thiết kế này, lớp 1 đóng vai trò như một "bộ lọc thô" giàu tính bao phủ: mọi bệnh nhân có khả năng khớp với ảnh truy vấn đều nằm trong danh sách ứng viên, trong khi chi phí tính toán vẫn rất thấp dù cơ sở dữ liệu có thể mở rộng lên hàng chục nghìn người.

3.2.4.2. Lớp lọc 2 – Fine Verification bằng ArcFace + cosine similarity

Lớp lọc thứ hai (Fine Verification) được thiết kế với mục tiêu ưu tiên độ chính xác tuyệt đối thay vì tốc độ xử lý. Sau khi lớp lọc Coarse Search trả về danh sách top-K ứng viên tiềm năng, lớp Fine Verification đảm nhiệm vai trò đưa ra quyết định định danh cuối cùng. Dựa trên kết quả phân tích thực nghiệm trình bày ở mục 3.2.3.2, hệ thống lựa chọn mô hình ArcFace (InsightFace) với embedding 512 chiều cho cả cơ sở dữ liệu và ảnh truy vấn, kết hợp với cosine similarity làm thước đo độ tương đồng.

Ngưỡng quyết định được thiết lập ở mức THRESHOLD = 0,60, tương ứng với mức độ tương đồng khoảng 80% khi biểu diễn trên thang phần trăm. Giá trị ngưỡng này được lựa chọn dựa trên đặc tính phân bố similarity của ArcFace trong thực nghiệm, đảm bảo không phát sinh trường hợp nhận diện nhầm trong tập dữ liệu thử nghiệm.

Quy trình xử lý tại lớp Fine Verification được thực hiện theo các bước sau:

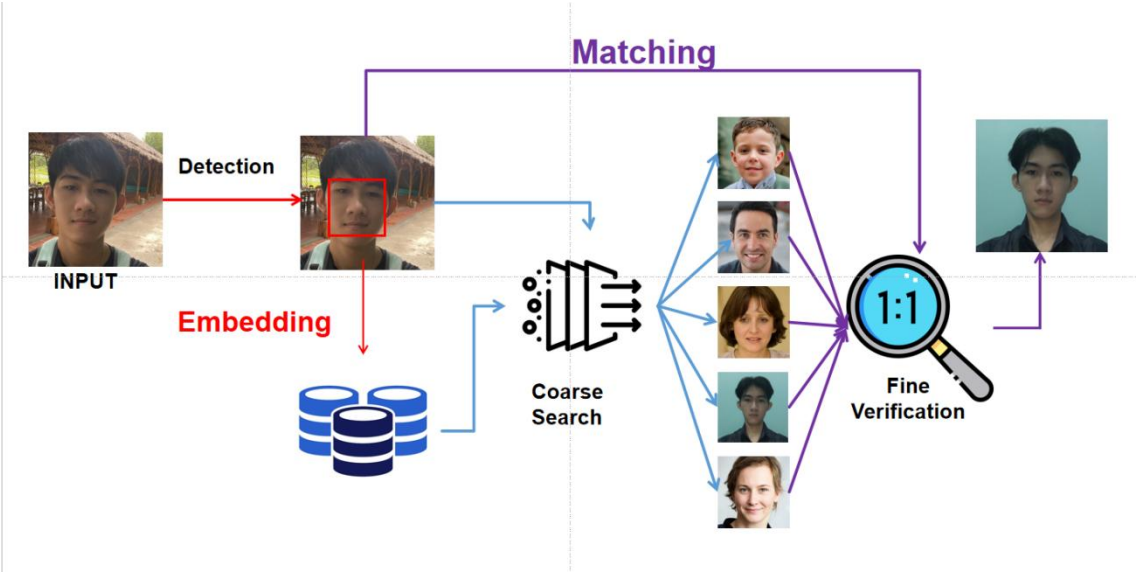
1. Trích xuất embedding ArcFace của ảnh truy vấn.
2. Lấy embedding tương ứng của từng ứng viên trong danh sách top-K từ lớp Coarse Search.
3. Tính toán độ tương đồng cosine giữa embedding truy vấn và embedding của từng ứng viên.
4. Lựa chọn ứng viên có giá trị cosine similarity cao nhất.
5. So sánh giá trị này với ngưỡng quyết định THRESHOLD = 0,60.

Trong trường hợp giá trị cosine similarity lớn hơn ngưỡng, hệ thống xác nhận danh tính bệnh nhân và trả về mã bệnh nhân tương ứng. Ngược lại, nếu giá trị này nhỏ hơn ngưỡng, hệ thống kết luận bệnh nhân chưa tồn tại trong cơ sở dữ liệu nhận diện và chuyển sang quy trình tiếp nhận bệnh nhân mới.

Kết quả thực nghiệm cho thấy, với mô hình ArcFace, phân bố cosine similarity được tách thành hai cụm rõ ràng. Cụm match có giá trị cao và ổn định, dao động trong khoảng 80–92% (tương đương cosine từ khoảng 0,60 đến 0,90), trong khi cụm non-match tập trung ở mức thấp hơn, khoảng 53–60%. Trên bình diện thống kê, hai phân bố này gần như không chồng lấn, với giá trị trung bình xấp xỉ 0,68 cho match và 0,11 cho non-match.

Sự tách biệt rõ ràng này giúp quá trình ra quyết định ở lớp Fine Verification trở nên đơn giản, nhất quán và an toàn, đồng thời giảm đáng kể rủi ro nhận diện nhầm. Đặc tính này đặc biệt quan trọng trong bối cảnh ứng dụng y tế, nơi một sai sót nhỏ trong định danh bệnh nhân có thể dẫn đến những hậu quả nghiêm trọng trong quá trình khám và điều trị.

3.2.4.3. Luồng xử lý tổng thể và lợi ích của cơ chế lọc hai lớp



Hình 3.6 Minh họa luồng xử lý tổng thể

Toàn bộ pipeline Detection → Embedding → Matching khi tích hợp cơ chế lọc hai lớp được hiện thực trong hàm check_face_in_db() của mô-đun Face ID, với luồng xử lý tóm tắt như sau:

1. **Tiền xử lý & phát hiện khuôn mặt**. Ảnh gốc được chuẩn hóa kích thước, chuyển đổi không gian màu và đưa vào mô hình YOLO-face để xác định bounding box có độ tin cậy cao nhất. Hệ thống cắt vùng mặt, bổ sung margin và chuyển sang khối Embedding.

2. **Sinh embedding ArcFace cho ảnh truy vấn**. Vùng mặt được đưa vào mô hình ArcFace (InsightFace – buffalo_l), trả về vector embedding 512D đã được chuẩn hóa.
3. **Lớp lọc 1 – Coarse Search**. Embedding truy vấn được FAISS IndexFlatL2 tìm kiếm trên toàn bộ cơ sở dữ liệu, trả về top-K ứng viên. Bước này đảm bảo tốc độ truy vấn cao với recall@10 = 100%.
4. **Lớp lọc 2 – Fine Verification**. Hệ thống tính cosine similarity chính xác giữa embedding truy vấn và K ứng viên, chọn ra ứng viên tốt nhất và so sánh với ngưỡng. Dựa trên kết quả này, hệ thống trả về:
   - Mã bệnh nhân và độ tin cậy nếu vượt ngưỡng;
   - Thông báo không tồn tại trong database nếu dưới ngưỡng.
5. **Khả năng mở rộng và cập nhật động**. Cơ sở dữ liệu embedding và index FAISS được xây dựng từ trước và có thể cập nhật khi thêm bệnh nhân mới mà không phải huấn luyện lại toàn bộ mô hình. Điều này giúp hệ thống dễ dàng mở rộng theo thời gian, phù hợp với các bệnh viện có số lượng bệnh nhân đăng ký mới mỗi ngày.

   Cơ chế lọc hai lớp như trên mang lại một số lợi ích quan trọng:

   - Hiệu năng cao: thời gian xử lý cho mỗi truy vấn ở mức mili-giây, đáp ứng yêu cầu vận hành thời gian thực tại kiosk và dashboard.
   - Độ tin cậy cao: decision cuối cùng dựa trên embedding ArcFace với biên phân tách lớn, giảm thiểu nguy cơ nhầm lẫn danh tính.
   - Tính mô-đun và linh hoạt: mỗi lớp (coarse/fine) có thể được tinh chỉnh hoặc thay thế độc lập (ví dụ thay IndexFlatL2 bằng IVF khi cơ sở dữ liệu đạt hàng trăm nghìn bản ghi), trong khi giao diện API với các mô-đun khác của hệ thống vẫn được giữ nguyên.

Như vậy, cơ chế lọc hai lớp không chỉ là bước hiện thực cụ thể cho khối Matching, mà còn là thành phần cốt lõi giúp mô-đun Face ID đạt được mục tiêu ban đầu: vừa nhanh, vừa chính xác, vừa an toàn khi tích hợp vào hệ thống quản lý hồ sơ bệnh nhân và xếp hàng khám bệnh.

3.3. Triển khai module bốc số và quản lý hàng đợi

Module bốc số và quản lý hàng đợi chịu trách nhiệm điều phối luồng bệnh nhân từ khi đăng ký dịch vụ tại kiosk đến khi được gọi vào phòng khám, thăm khám và kết thúc phiên. Module này phải đáp ứng đồng thời các yêu cầu:

- Hỗ trợ nhiều dịch vụ, nhiều phòng khám, nhiều bác sĩ hoạt động song song;
- Đảm bảo tính công bằng (first-come-first-served), nhưng vẫn có thể mở rộng cho các ưu tiên sau này (cấp cứu, lịch hẹn trước…);

- Cập nhật trạng thái thời gian gần thực tới dashboard và màn hình hiển thị số.

Để làm được điều đó, cần xây dựng một mô hình hàng đợi rõ ràng, cấu trúc dữ liệu phù hợp và bộ thuật toán điều phối hai lớp.

3.3.1. Mô hình hàng đợi và luồng xử lý cơ bản

Mô hình hàng đợi trong hệ thống được thiết kế nhằm mô tả đầy đủ luồng xử lý của một bệnh nhân kể từ thời điểm đăng ký khám tại kiosk cho đến khi hoàn thành phiên khám tại phòng khám. Toàn bộ quy trình được tổ chức theo các bước liên tiếp, đảm bảo tính công bằng, nhất quán và khả năng cập nhật theo thời gian thực giữa các thành phần của hệ thống.

Trước hết, bệnh nhân thực hiện đăng ký dịch vụ hoặc chuyên khoa tại kiosk. Tại đây, bệnh nhân có thể là người mới hoặc đã được nhận diện thông qua mô-đun Face ID. Người dùng tiến hành xác nhận thông tin cá nhân, lựa chọn một hoặc nhiều dịch vụ khám phù hợp với nhu cầu như Nội tổng quát, Nhi, Tim mạch hoặc Chẩn đoán hình ảnh, và trong một số trường hợp có thể chọn bác sĩ ưu tiên. Sau khi hoàn tất thao tác, kiosk gửi yêu cầu đăng ký lên server bốc số, bao gồm mã bệnh nhân (patient_id), danh sách mã dịch vụ (service_id dạng DVxxx) và thông tin bác sĩ ưu tiên (nếu có).

Khi tiếp nhận yêu cầu đăng ký, module bốc số tiến hành sinh số thứ tự và đưa bệnh nhân vào hàng đợi tạm thời. Hệ thống tra cứu bảng service để xác định chuyên khoa và nhóm phòng khám tương ứng cho từng dịch vụ, đồng thời kết hợp với các bảng room và doctor_information để xác định những phòng và bác sĩ có khả năng tiếp nhận. Số thứ tự (ticket_number) được sinh theo quy tắc riêng cho từng khu vực hoặc dịch vụ, chẳng hạn như Nxxx cho Nội tổng quát, TMxxx cho Tim mạch hoặc CDHAxxx cho Chẩn đoán hình ảnh. Mỗi dịch vụ mà bệnh nhân đăng ký sẽ tạo ra một nhu cầu khám tương ứng và được ghi nhận trong cấu trúc hàng đợi tạm thời (wait_queue). Ở giai đoạn này, bệnh nhân chỉ được "xếp chỗ" trong các hàng đợi phù hợp, chưa bị gán cố định cho một phòng hay bác sĩ cụ thể.

Trong suốt thời gian chờ khám, mỗi phiên khám của bệnh nhân được gắn với một trạng thái để phục vụ theo dõi và điều phối. Các trạng thái chính bao gồm:

- WAITING: bệnh nhân đang chờ đến lượt khám;
- CALLING: bệnh nhân đang được gọi vào phòng khám;
- SERVING: bệnh nhân đang trong quá trình khám;
- DONE: phiên khám đã hoàn thành;
- SKIPPED: bệnh nhân không có mặt khi được gọi.

Các trạng thái này được hiển thị trên dashboard điều phối và màn hình hiển thị số trước các phòng khám, cho phép nhân viên lễ tân, bác sĩ và bệnh nhân theo dõi số thứ

tự đang được phục vụ, số tiếp theo sẽ được gọi và danh sách bệnh nhân đang chờ tại từng khu vực.

Khi bác sĩ thao tác trên giao diện phòng khám, chẳng hạn như gọi bệnh nhân tiếp theo, bắt đầu khám hoặc kết thúc khám, module hàng đợi sẽ cập nhật trạng thái tương ứng trong các bảng real_time và reception_log. Các lượt không có mặt khi được gọi sẽ được đánh dấu là SKIPPED, đồng thời hệ thống ghi nhận thời điểm thay đổi trạng thái như thời gian gọi, thời gian bắt đầu và kết thúc khám. Những thay đổi này được đồng bộ gần như tức thời lên dashboard điều phối và màn hình hiển thị số, giúp toàn bộ hệ thống duy trì một "nguồn dữ liệu thống nhất" (single source of truth) về tình trạng xếp hàng.

Nhờ cơ chế đồng bộ này, các thành phần như kiosk, dashboard điều phối, giao diện phòng khám và mô-đun Face ID đều hoạt động trên cùng một bộ dữ liệu hàng đợi. Điều này giúp giảm thiểu xung đột như trùng số thứ tự hoặc trùng lượt khám khi nhiều người thao tác đồng thời, đồng thời tạo điều kiện thuận lợi để mở rộng hệ thống trong tương lai, chẳng hạn như bổ sung các luật ưu tiên hoặc các loại dịch vụ mới mà không cần thay đổi kiến trúc tổng thể.

3.3.2. Cấu trúc dữ liệu dùng để quản lý hàng đợi

Module bốc số được hiện thực dựa trên các bảng và cấu trúc dữ liệu đã được thiết kế ở Mục 2.6 – Thiết kế cơ sở dữ liệu. Trong đó, các bảng liên quan trực tiếp đến chức năng xếp hàng và điều phối bệnh nhân được chia thành ba nhóm chính, bao gồm nhóm danh mục cấu hình, nhóm bảng hàng đợi và trạng thái thời gian thực, và nhóm bảng nhật ký phục vụ theo dõi và báo cáo.

Nhóm danh mục cấu hình đóng vai trò nền tảng cho thuật toán bốc số và điều phối. Bảng service (Bảng 2.1) lưu trữ danh mục các dịch vụ khám bệnh, với các trường quan trọng như service_id, service_name, specialty, room_id và prioritize. Bảng này giữ vai trò là "điểm nối" giữa yêu cầu đăng ký của bệnh nhân (mã dịch vụ DVxxx do kiosk gửi lên) với chuyên khoa và phòng khám tương ứng. Đồng thời, cờ prioritize cho phép hệ thống nhận biết các dịch vụ có mức độ ưu tiên, tạo tiền đề để mở rộng các hàng đợi ưu tiên như cấp cứu, bệnh nhân VIP hoặc lịch hẹn trước trong tương lai.

Bảng room (Bảng 2.2) lưu trữ thông tin danh mục phòng khám, với các trường chính gồm room_id, room_name, room_type và active. Bảng này cung cấp danh sách các phòng khám đang hoạt động để thuật toán điều phối có thể phân bổ lượt khám phù hợp. Trường active cho phép tạm thời vô hiệu hóa một phòng khám, chẳng hạn trong trường hợp bảo trì hoặc nghỉ ca, mà không làm ảnh hưởng đến hoạt động chung của toàn bộ hệ thống. Bên cạnh đó, bảng doctor_information (Bảng 2.3) lưu trữ thông tin bác sĩ với các trường như doctor_id, doctor_name, specialty và room_id. Đây là cơ sở để hệ thống gợi ý bác sĩ phù hợp với từng dịch vụ hoặc chuyên khoa, đồng thời trường

room_id cho phép gắn bác sĩ với phòng khám chính, phục vụ cho lớp điều phối thứ hai khi gán lượt khám cụ thể cho bác sĩ và phòng.

Nhóm bảng hàng đợi và trạng thái thời gian thực là trung tâm của module bốc số. Bảng wait_queue (Bảng 2.4) được thiết kế dưới dạng bảng phẳng, trong đó mỗi dòng (wait_index) tương ứng với một "lượt chờ" tổng quát, còn các cột P001, P002,… đại diện cho các phòng khám. Quy ước quan trọng của bảng này là mỗi patient_id chỉ được phép xuất hiện ở một cột duy nhất trong cùng một dòng, đảm bảo mô hình "một bệnh nhân – một vị trí". Trong trường hợp bệnh nhân đăng ký nhiều dịch vụ thuộc các phòng khác nhau, thuật toán sẽ sắp xếp sao cho quy tắc này vẫn được duy trì, tránh trùng lặp bệnh nhân trong cùng một hàng đợi. Bảng wait_queue có vai trò lưu trữ các lượt chờ tạm thời sau khi bốc số, trước khi bệnh nhân được điều phối sang hàng đợi thực tế.

Bảng real_time (Bảng 2.5) có cấu trúc tương tự wait_queue, nhưng mỗi dòng phản ánh trạng thái hàng chờ thực tế trước cửa các phòng khám tại một thời điểm cụ thể. Bảng này là nguồn dữ liệu chính cho dashboard điều phối và các màn hình hiển thị số, thể hiện danh sách các bệnh nhân sắp được gọi hoặc đang trực tiếp chờ khám tại từng phòng.

Nhóm bảng nhật ký và báo cáo được đại diện bởi bảng reception_log (Bảng 2.6). Bảng này lưu trữ thông tin lịch sử của mỗi lượt tiếp nhận, với các trường quan trọng như patient_id, reception_time, service và status. Dữ liệu trong reception_log cho phép hệ thống theo dõi toàn bộ quá trình của một lượt khám, từ thời điểm đăng ký đến khi hoàn thành, đồng thời là nguồn dữ liệu phục vụ thống kê và báo cáo, chẳng hạn như số lượt khám theo ngày, theo dịch vụ, theo bác sĩ hoặc thời gian chờ trung bình của bệnh nhân.

3.3.3. Quy trình thiết kế và tối ưu thuật toán bốc số – điều phối bệnh nhân

Quy trình thiết kế thuật toán bốc số và điều phối bệnh nhân được thực hiện dựa trên việc phân tích kỹ các yêu cầu nghiệp vụ và các ràng buộc của hệ thống trong môi trường bệnh viện thực tế. Trong bối cảnh bệnh viện có nhiều khu vực khám khác nhau, mỗi khu vực bao gồm nhiều phòng và mỗi phòng có thể có một hoặc nhiều bác sĩ làm việc theo ca, hệ thống cần xử lý linh hoạt các tình huống phát sinh trong quá trình tiếp nhận. Đồng thời, một bệnh nhân có thể đăng ký nhiều dịch vụ trong cùng một lần tiếp nhận, chẳng hạn như khám nội tổng quát kết hợp siêu âm hoặc xét nghiệm. Do đó, thuật toán phải đảm bảo nguyên tắc "đến trước – phục vụ trước" trong phạm vi từng loại dịch vụ, nhưng vẫn cho phép mở rộng các quy tắc ưu tiên trong tương lai. Ngoài ra, dữ liệu và thuật toán cần được thiết kế gọn nhẹ để có thể xử lý nhanh trên một server cấu hình vừa phải, đồng thời dễ tích hợp với mô-đun Face ID và các thành phần khác của hệ thống.

Từ các ràng buộc trên, đã lựa chọn chiến lược bốc số theo hướng tách biệt giữa việc xếp hàng theo dịch vụ và việc gán phòng khám cụ thể. Theo đó, mỗi dịch vụ được gắn với một nhóm phòng khám thuộc cùng chuyên khoa. Khi bệnh nhân đăng ký, hệ thống không gán ngay bệnh nhân vào một phòng cố định mà đưa yêu cầu vào hàng đợi tạm thời (wait_queue), cho phép việc điều phối giữa các phòng trong cùng chuyên khoa được thực hiện linh hoạt ở các bước sau. Số thứ tự được sinh riêng cho từng khu vực hoặc từng nhóm dịch vụ, giúp tránh xung đột và hỗ trợ bệnh nhân dễ nhận biết thứ tự khám của mình.

Để hiện thực hóa chiến lược trên, hệ thống được thiết kế với hai lớp điều phối độc lập. Lớp điều phối thứ nhất chịu trách nhiệm phân bổ bệnh nhân vào các vị trí trong wait_queue dựa trên dịch vụ đăng ký và chuyên khoa tương ứng. Lớp điều phối thứ hai đọc dữ liệu từ wait_queue và real_time, sau đó gán bệnh nhân cụ thể vào phòng khám hoặc bác sĩ đang rảnh, hoặc có mức tải thấp hơn. Việc tách biệt rõ ràng hai lớp điều phối giúp thuật toán dễ bảo trì và dễ mở rộng, cho phép bổ sung thêm phòng khám, bác sĩ hoặc các luật ưu tiên mới mà không cần thay đổi toàn bộ logic xử lý.

Trong quá trình triển khai, thuật toán được tối ưu hóa và kiểm thử qua nhiều vòng lặp. Hệ thống được thử nghiệm với các tập dữ liệu mô phỏng gồm nhiều bệnh nhân, nhiều dịch vụ và nhiều phòng khám nhằm đánh giá thời gian sinh số thứ tự, thời gian cập nhật các bảng wait_queue và real_time, cũng như khả năng xử lý khi nhiều kiosk đồng thời gửi yêu cầu. Đồng thời, các quy tắc quan trọng được rà soát để đảm bảo không xảy ra trường hợp một mã bệnh nhân xuất hiện ở nhiều cột trong cùng một hàng đợi, và các slot trong wait_queue được giải phóng hợp lý sau khi bệnh nhân đã hoàn thành lượt khám. Tính công bằng của thuật toán cũng được kiểm tra nhằm bảo đảm bệnh nhân đến trước luôn được xếp trước trong cùng một hàng đợi dịch vụ, trừ khi có áp dụng các luật ưu tiên đặc biệt.

Kết quả kiểm thử cho thấy mô hình điều phối hai lớp giúp hệ thống duy trì được tính đơn giản trong triển khai, đồng thời mang lại sự linh hoạt cao khi cấu hình thêm dịch vụ hoặc phòng khám mới. Đây là cơ sở quan trọng để hệ thống có thể mở rộng và thích ứng với các kịch bản vận hành khác nhau trong môi trường bệnh viện thực tế.

3.3.4. Lớp điều phối 1 (theo dịch vụ/phòng khám) và lớp điều phối 2 (theo bác sĩ/trạng thái phòng)

Hệ thống điều phối bệnh nhân được thiết kế theo mô hình hai lớp độc lập nhằm tách biệt rõ ràng giữa quá trình bốc số – xếp hàng ban đầu và quá trình gán bệnh nhân vào bác sĩ, phòng khám cụ thể. Cách tiếp cận này giúp hệ thống vừa đảm bảo tính công bằng trong xếp hàng, vừa duy trì sự linh hoạt khi điều phối theo trạng thái thực tế của phòng khám và bác sĩ.

Lớp điều phối 1 được kích hoạt ngay khi kiosk gửi yêu cầu đăng ký khám lên server và đóng vai trò xây dựng hàng đợi tạm thời (wait_queue). Đầu vào của lớp này bao gồm mã bệnh nhân (patient_id), danh sách dịch vụ đăng ký (service_id) và thông tin bác sĩ ưu tiên (nếu có). Với mỗi dịch vụ, hệ thống tra cứu bảng cấu hình dịch vụ để xác định chuyên khoa tương ứng, phòng khám mặc định và các cờ ưu tiên, đồng thời kết hợp với bảng phòng khám để xác định các phòng đang hoạt động thuộc cùng chuyên khoa.

Dựa trên các thông tin này, thuật toán tiến hành phân bổ bệnh nhân vào wait_queue. Cụ thể, hệ thống tìm dòng (wait_index) đầu tiên mà cột phòng tương ứng còn trống và dòng đó chưa chứa mã bệnh nhân ở bất kỳ cột nào khác. Nếu tìm được vị trí phù hợp, bệnh nhân được đưa vào đúng cột phòng tương ứng; trong trường hợp không còn dòng trống thỏa mãn, hệ thống sẽ tự động sinh thêm một dòng mới và gán bệnh nhân vào đó. Mỗi lần bệnh nhân được xếp vào hàng đợi, hệ thống sinh số thứ tự theo khu vực hoặc dịch vụ, ghi nhận vào bảng reception_log kèm theo thời gian tiếp nhận và danh sách dịch vụ đăng ký, với trạng thái ban đầu là WAITING. Sau khi lớp điều phối 1 hoàn tất, tất cả yêu cầu đăng ký đều được xếp hàng một cách nhất quán và công bằng, nhưng chưa bị gán cố định cho một bác sĩ cụ thể.

Lớp điều phối 2 chịu trách nhiệm điều phối bệnh nhân vào hàng đợi thực tế (real_time) dựa trên trạng thái vận hành của phòng khám và bác sĩ. Lớp này được kích hoạt khi một phiên khám kết thúc và phòng được giải phóng, hoặc khi hệ thống thực hiện kiểm tra trạng thái định kỳ. Trước hết, hệ thống đọc dữ liệu từ bảng real_time và room để xác định các phòng đang rảnh hoặc có thể tiếp nhận thêm bệnh nhân, đồng thời kiểm tra trạng thái làm việc của bác sĩ gắn với từng phòng thông qua bảng doctor_information.

Khi phát hiện phòng phù hợp, hệ thống tiến hành chọn bệnh nhân từ wait_queue theo thứ tự wait_index tăng dần, đảm bảo nguyên tắc "đến trước – phục vụ trước". Đối với mỗi phòng, hệ thống tìm dòng đầu tiên mà cột phòng tương ứng chứa mã bệnh nhân, sau đó kiểm tra thêm các tiêu chí phụ như sự phù hợp chuyên khoa hoặc việc bệnh nhân có chọn bác sĩ ưu tiên đang làm việc tại phòng đó hay không. Khi bệnh nhân được chọn, hệ thống chuyển thông tin sang real_time, cập nhật trạng thái lượt khám sang CALLING hoặc SERVING tùy thời điểm.

Sau khi quá trình chuyển bệnh nhân hoàn tất, slot tương ứng trong wait_queue được giải phóng hoặc đánh dấu đã xử lý. Đồng thời, bảng reception_log được cập nhật trạng thái mới như SERVING, DONE hoặc SKIPPED theo thao tác của bác sĩ. Toàn bộ thay đổi này được đồng bộ gần như tức thời lên dashboard điều phối và màn hình hiển thị số, đảm bảo nhân viên lễ tân, bác sĩ và bệnh nhân đều quan sát được thứ tự khám mới nhất.

Việc tách hệ thống điều phối thành hai lớp mang lại nhiều lợi ích quan trọng. Thứ nhất, hệ thống đạt được tính linh hoạt cao khi có thể thay đổi logic gán phòng hoặc bác sĩ ở lớp điều phối 2 mà không ảnh hưởng đến cơ chế bốc số cơ bản của lớp điều phối 1. Thứ hai, nguyên tắc công bằng "first-come-first-served" vẫn được bảo toàn trong phạm vi từng dịch vụ và phòng khám. Cuối cùng, kiến trúc này tạo điều kiện thuận lợi cho việc mở rộng hệ thống, cho phép bổ sung các luật ưu tiên, thêm phòng khám, thêm bác sĩ hoặc triển khai nhiều kiosk song song chỉ bằng cách điều chỉnh thuật toán ở lớp điều phối thứ hai.

3.4. Giao diện người dùng

Trong khuôn khổ đề tài "Nghiên cứu và xây dựng API định danh bệnh nhân bằng nhận diện khuôn mặt và điều phối hàng đợi thời gian thực", hệ thống được triển khai dưới dạng ứng dụng web, đóng vai trò là lớp giao diện tương tác giữa người dùng và các API phía máy chủ. Phần giao diện người dùng được xây dựng trên nền tảng Vue.js 3, trong khi phía máy chủ sử dụng FastAPI để cung cấp các API nhận diện khuôn mặt và điều phối hàng đợi theo thời gian thực.

Mặc dù trọng tâm của đề tài là nghiên cứu và xây dựng các API, việc thiết kế giao diện người dùng có ý nghĩa quan trọng trong việc minh họa khả năng vận hành thực tế của hệ thống, đồng thời hỗ trợ đánh giá tính khả thi của các API trong môi trường bệnh viện. Giao diện giúp bệnh nhân thực hiện các thao tác định danh và đăng ký khám một cách nhanh chóng, đồng thời hỗ trợ nhân viên y tế theo dõi và điều phối luồng bệnh nhân theo thời gian thực.

Trong phạm vi nghiên cứu, hệ thống tập trung xây dựng ba nhóm giao diện người dùng chính, tương ứng với các kịch bản sử dụng điển hình của các API đã thiết kế. Thứ nhất là giao diện nhận diện khuôn mặt, phục vụ việc thu nhận ảnh khuôn mặt và gửi yêu cầu nhận diện tới API định danh bệnh nhân. Thứ hai là giao diện kiosk đăng ký khám, cho phép bệnh nhân lựa chọn dịch vụ khám và nhận số thứ tự thông qua API điều phối hàng đợi. Thứ ba là giao diện dashboard điều phối, cung cấp cho nhân viên y tế và bộ phận quản lý cái nhìn tổng thể về trạng thái hàng đợi, tình hình tiếp nhận bệnh nhân và trạng thái hoạt động của các phòng khám theo thời gian thực.

Việc phân tách giao diện theo từng nhóm chức năng và đối tượng sử dụng không chỉ giúp hệ thống dễ dàng triển khai trong môi trường bệnh viện thực tế, mà còn thể hiện rõ vai trò trung tâm của các API định danh khuôn mặt và API điều phối hàng đợi thời gian thực trong kiến trúc tổng thể của hệ thống.

3.4.1. Giao diện kiosk: lựa chọn dịch vụ.

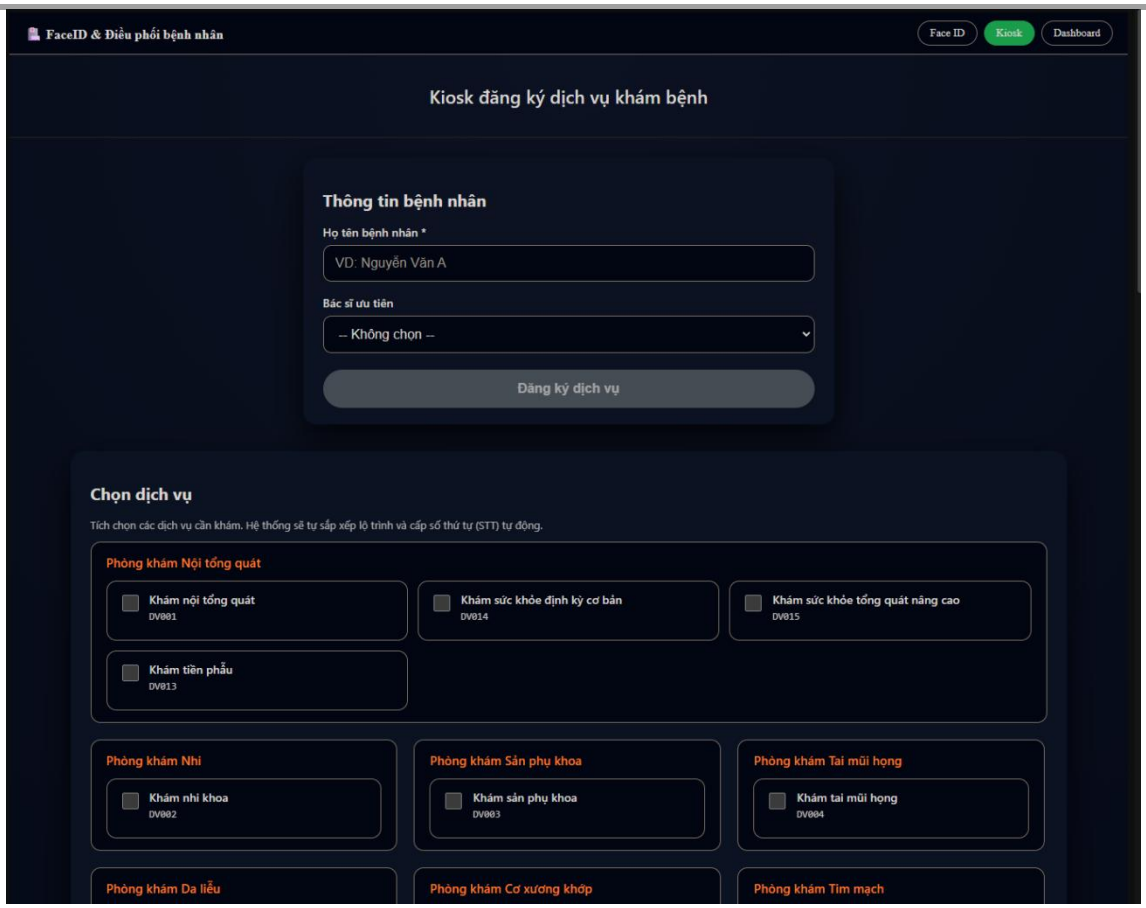
Giao diện Kiosk là trang chức năng dành cho bệnh nhân sử dụng trực tiếp tại hệ thống bốc số tự động của bệnh viện. Mục tiêu chính của giao diện này là hỗ trợ bệnh nhân thực hiện các thao tác đăng ký khám bệnh một cách đơn giản, nhanh chóng và

hạn chế tối đa việc nhập liệu thủ công, qua đó giảm áp lực cho khu vực tiếp nhận và nâng cao trải nghiệm người dùng.

Giao diện được xây dựng trên tệp Kiosk.vue với bố cục trực quan, tối ưu cho màn hình cảm ứng. Nội dung giao diện được chia thành hai phần chính. Phần thứ nhất cho phép bệnh nhân nhập thông tin cá nhân cần thiết. Trong trường hợp bệnh nhân đến khám lần đầu, hệ thống yêu cầu nhập họ tên để phục vụ việc tạo hồ sơ tạm thời. Ngoài ra, bệnh nhân có thể lựa chọn bác sĩ ưu tiên thông qua danh sách thả xuống. Thông tin này được liên kết trực tiếp với cơ sở dữ liệu của hệ thống nhằm hỗ trợ quá trình phân phòng và điều phối bệnh nhân chính xác hơn.

Phần thứ hai của giao diện là khu vực lựa chọn dịch vụ khám bệnh. Các dịch vụ được hiển thị dưới dạng các ô lựa chọn (checkbox) và được nhóm theo chuyên khoa, giúp bệnh nhân dễ dàng tìm kiếm và lựa chọn đúng nhu cầu khám. Mỗi dịch vụ được gán một mã định danh riêng, chẳng hạn như DV001, DV006 hoặc DV802, nhằm hỗ trợ backend xử lý dữ liệu và thực hiện thuật toán điều phối. Sau khi hoàn tất việc lựa chọn dịch vụ, bệnh nhân nhấn nút “Xác nhận đăng ký” để gửi yêu cầu bốc số vào hệ thống hàng đợi.

Giao diện kiosk được thiết kế với kích thước chữ lớn, màu sắc rõ ràng và thao tác đơn giản, phù hợp với môi trường bệnh viện và đối tượng sử dụng đa dạng, bao gồm cả người cao tuổi. Thiết kế này giúp giảm sai sót trong quá trình đăng ký và rút ngắn thời gian thao tác tại kiosk.



Hình 3.7 Minh họa kiểm thử module Kiosk đăng ký khám

Hình 3.7 minh họa giao diện kiểm thử của mô-đun Kiosk đăng ký khám. Hình ảnh thể hiện quá trình bệnh nhân nhập thông tin cá nhân và lựa chọn dịch vụ trực tiếp trên giao diện. Sau khi xác nhận đăng ký, dữ liệu được gửi đến server để xử lý và đưa bệnh nhân vào hàng đợi tương ứng theo thuật toán điều phối của hệ thống.

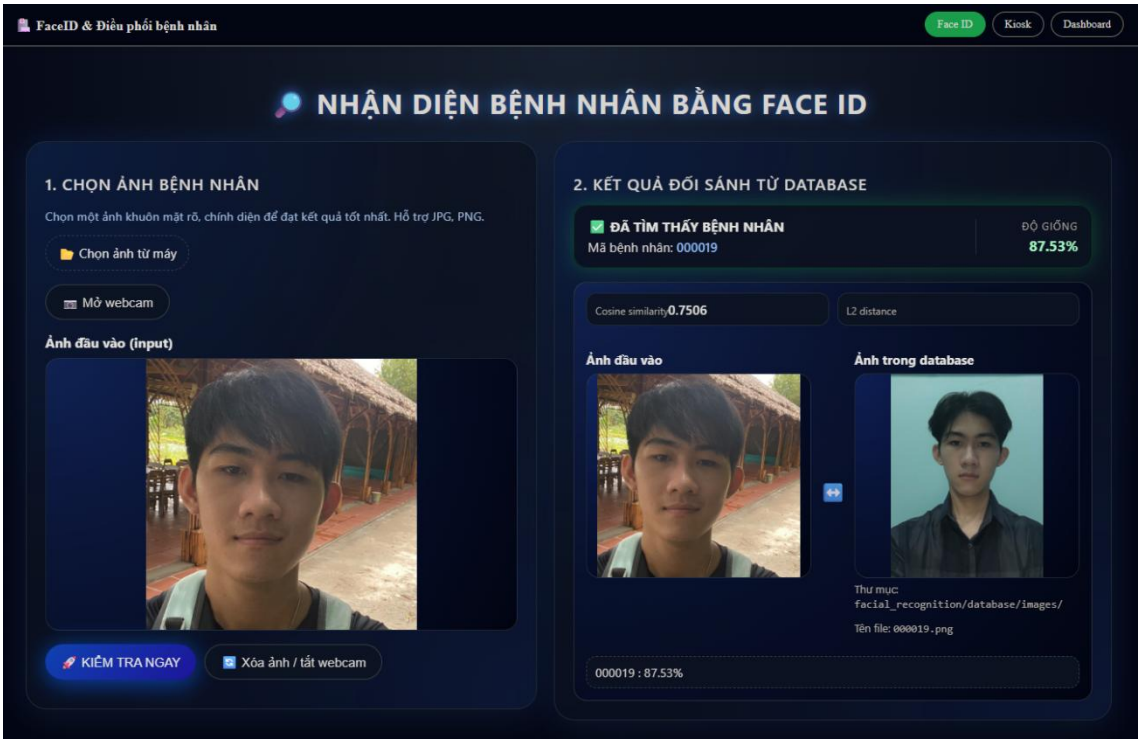
3.4.2. Giao diện module Face ID

Giao diện Face ID được xây dựng như một trang chức năng riêng biệt nhằm phục vụ việc thử nghiệm, đánh giá và kiểm thử thuật toán nhận diện khuôn mặt được tích hợp trong hệ thống. Đây không phải là giao diện chính của kiosk tiếp nhận bệnh nhân, mà đóng vai trò như một công cụ kiểm chứng kỹ thuật, giúp đánh giá hiệu quả và độ chính xác của mô-đun nhận diện trước khi tích hợp hoàn chỉnh vào quy trình bốc số và điều phối khám bệnh. Giao diện này được hiện thực hóa thông qua tệp Facial.vue trong hệ thống frontend.

Trang Face ID cho phép người dùng cung cấp dữ liệu đầu vào thông qua hai hình thức chính. Người dùng có thể lựa chọn ảnh chân dung trực tiếp từ máy tính thông qua nút “Chọn ảnh từ máy”, hoặc sử dụng nút “Mở webcam” để hiển thị hình ảnh video thời gian thực từ camera. Khi người dùng nhấn nút “Kiểm tra ngay”, hệ thống sẽ tự động chụp khung hình hiện tại (đối với webcam) hoặc sử dụng ảnh đã chọn, sau đó tiến hành xử lý nhận diện.

Ảnh đầu vào sau khi thu nhận được đưa vào mô hình InsightFace để thực hiện trích xuất vector đặc trưng (embedding). Các embedding này được so sánh với cơ sở dữ liệu khuôn mặt của bệnh nhân nhằm xác định mức độ tương đồng. Kết quả nhận diện được hiển thị trực quan trên giao diện, bao gồm các thông tin như mã bệnh nhân tìm thấy (nếu tồn tại), tỷ lệ giống nhau dưới dạng phần trăm, các chỉ số đánh giá như cosine similarity và khoảng cách L2, đồng thời hiển thị song song ảnh đầu vào và ảnh tương ứng trong cơ sở dữ liệu để người dùng dễ dàng đối chiếu.

Về mục đích sử dụng, giao diện Face ID đóng vai trò quan trọng trong quá trình kiểm thử và đánh giá thuật toán nhận diện khuôn mặt. Thông qua việc quan sát trực tiếp kết quả so khớp và các chỉ số định lượng, hệ thống cho phép đánh giá chất lượng mô-đun nhận diện trong nhiều điều kiện khác nhau. Đây cũng là cơ sở để tinh chỉnh tham số, kiểm tra độ ổn định và xác nhận mức độ sẵn sàng của mô-đun Face ID trước khi tích hợp hoàn toàn vào hệ thống bốc số và điều phối khám bệnh.



Hình 3.8 Minh họa kiểm thử module Face ID

Hình 3.8 minh họa giao diện thử nghiệm của mô-đun Face ID trong hệ thống. Giao diện này cho phép kiểm tra toàn bộ quy trình nhận diện khuôn mặt, từ khâu thu nhận ảnh đầu vào đến bước so khớp và hiển thị kết quả nhận diện. Người dùng có thể lựa chọn ảnh khuôn mặt từ thiết bị hoặc sử dụng webcam để thu nhận hình ảnh theo thời gian thực, qua đó mô phỏng các tình huống sử dụng khác nhau trong môi trường thực tế.

3.4.3. Giao diện dashboard điều phối bệnh nhân

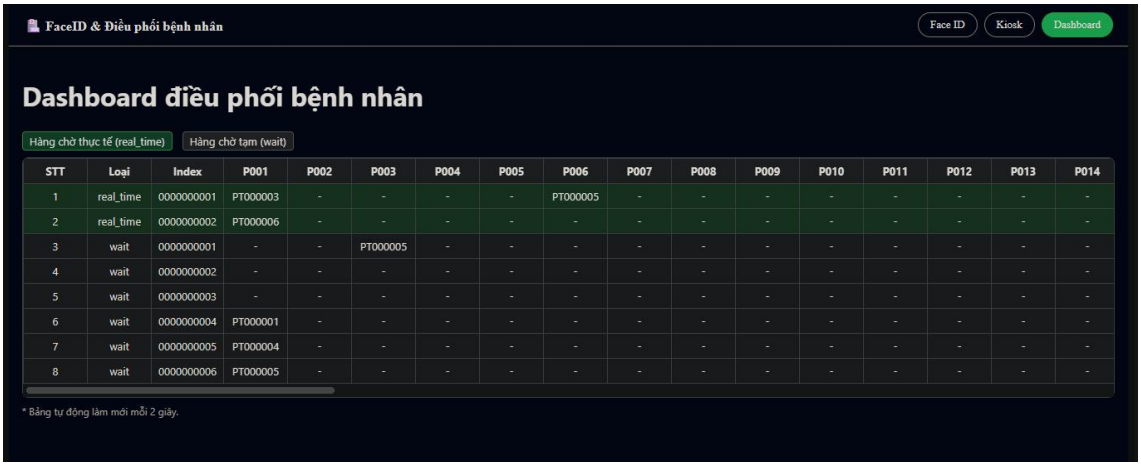
Giao diện Dashboard là trang chức năng dành cho nhân viên điều phối, cho phép theo dõi toàn bộ tình trạng khám chữa bệnh của hệ thống theo thời gian thực. Giao diện này được xây dựng thông qua tệp Dashboard.vue và đóng vai trò trung tâm trong việc giám sát hoạt động của các phòng khám, bác sĩ và luồng bệnh nhân trong suốt quá trình vận hành.

Trang Dashboard hiển thị một bảng dữ liệu tổng hợp có kích thước lớn, trong đó thông tin được cập nhật tự động theo chu kỳ nhằm đảm bảo độ chính xác và kịp thời. Nội dung hiển thị chính bao gồm hàng chờ thực tế (real_time), phản ánh danh sách các bệnh nhân đang trực tiếp chờ khám tại từng phòng. Mỗi hàng dữ liệu thể hiện rõ các phòng khám từ P001 đến P014 cùng với mã bệnh nhân tương ứng, đồng thời được thiết kế với màu sắc nổi bật để phân biệt với hàng chờ tạm.

Bên cạnh đó, dashboard còn hiển thị hàng chờ tạm (wait), đóng vai trò là lớp dữ liệu dự phòng phục vụ cho quá trình phân luồng linh hoạt. Dữ liệu trong hàng chờ tạm được sinh ra từ thuật toán điều phối của hệ thống, dựa trên các yếu tố như chuyên khoa đăng ký, tình trạng sẵn có của phòng khám, bác sĩ đang trực và trạng thái khám trước đó của bệnh nhân. Cách tổ chức này giúp hệ thống chủ động điều phối khi có thay đổi về tải hoặc trạng thái phòng khám.

Một đặc điểm quan trọng của dashboard là mô hình "một bệnh nhân – một vị trí", theo đó mỗi bệnh nhân chỉ xuất hiện duy nhất một lần trong mỗi hàng chờ. Các cột phòng khám được trình bày rõ ràng, cho phép nhân viên điều phối dễ dàng nhận biết bệnh nhân đang chờ tại phòng nào, từ đó hỗ trợ việc theo dõi và xử lý tình huống phát sinh một cách nhanh chóng.

Về cơ chế cập nhật, dashboard đọc dữ liệu trực tiếp từ các bảng liên quan trong cơ sở dữ liệu MySQL và áp dụng cơ chế tự động làm mới theo chu kỳ khoảng 2 giây. Nhờ đó, mọi thay đổi về trạng thái khám bệnh đều được phản ánh gần như tức thời trên giao diện, đáp ứng yêu cầu theo dõi và điều phối trong môi trường bệnh viện thực tế.



STT	Loại	Index	P001	P002	P003	P004	P005	P006	P007	P008	P009	P010	P011	P012	P013	P014
1	real_time	0000000001	PT000003	-	-	-	-	PT000005	-	-	-	-	-	-	-	-
2	real_time	0000000002	PT000006	-	-	-	-	-	-	-	-	-	-	-	-	-
3	wait	0000000001	-	-	PT000005	-	-	-	-	-	-	-	-	-	-	-
4	wait	0000000002	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	wait	0000000003	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	wait	0000000004	PT000001	-	-	-	-	-	-	-	-	-	-	-	-	-
7	wait	0000000005	PT000004	-	-	-	-	-	-	-	-	-	-	-	-	-
8	wait	0000000006	PT000005	-	-	-	-	-	-	-	-	-	-	-	-	-

Hình 3.9 Minh họa kiểm thử module Dashboard

Hình 3.9 minh họa giao diện kiểm thử của mô-đun Dashboard điều phối bệnh nhân. Hình ảnh cho thấy cách hệ thống hiển thị đồng thời hàng chờ thực tế và hàng chờ tạm, cũng như cơ chế cập nhật dữ liệu theo thời gian thực, qua đó giúp đánh giá trực quan hiệu quả của mô-đun dashboard trong quá trình vận hành hệ thống.

3.5. Một số vấn đề kỹ thuật và bảo mật

Bên cạnh việc xây dựng mô-đun bốc số và quản lý hàng đợi, hệ thống còn phải đáp ứng các yêu cầu về bảo mật, phân quyền và hiệu năng. Đặc thù của môi trường y tế là dữ liệu bệnh nhân thuộc nhóm thông tin nhạy cảm, đồng thời khối lượng truy cập có thể tăng đột biến vào các khung giờ cao điểm. Do đó, mô-đun cần được thiết kế theo hướng an toàn – tin cậy – mở rộng tốt. Mục này trình bày các vấn đề kỹ thuật quan trọng được cân nhắc khi triển khai hệ thống.

3.5.1. Cơ chế phân quyền và xác thực người dùng

Để đảm bảo hệ thống vận hành an toàn và dữ liệu không bị truy cập trái phép, luận văn áp dụng mô hình phân quyền dựa trên vai trò (Role-Based Access Control – RBAC). Mỗi nhóm người dùng được gán một tập quyền hạn rõ ràng, tương ứng với chức năng và trách nhiệm trong quy trình tiếp nhận và khám bệnh.

Nhóm người dùng kiosk, chủ yếu là bệnh nhân hoặc khách vãng lai, không cần đăng nhập và chỉ được phép gửi yêu cầu đăng ký dịch vụ để bốc số. Nhóm này không có quyền truy cập thông tin bệnh nhân khác hay các API mang tính quản trị. Việc nhận diện Face ID chỉ được thực hiện trên server và kiosk không được cấp quyền truy cập sâu vào dữ liệu nhận dạng.

Nhân viên lễ tân đăng nhập bằng tài khoản hệ thống và có quyền xem danh sách bệnh nhân đang chờ theo từng phòng, tra cứu nhật ký tiếp nhận và hồ sơ bệnh nhân. Tuy nhiên, nhóm này không được phép chỉnh sửa hoặc xóa dữ liệu hàng đợi cũng như không được tự ý thay đổi trạng thái phòng khám.

Bác sĩ và kỹ thuật viên phòng khám đăng nhập qua giao diện chuyên biệt, được phép gọi bệnh nhân, bắt đầu và kết thúc phiên khám, đồng thời cập nhật các trạng thái như CALLING, SERVING, DONE hoặc SKIPPED. Mỗi bác sĩ chỉ có thể xem danh sách bệnh nhân thuộc phòng khám của mình, đảm bảo nguyên tắc phân tách dữ liệu theo phạm vi làm việc.

Quản trị viên hệ thống (Admin) có toàn quyền cấu hình và quản lý hệ thống, bao gồm dịch vụ, phòng khám, bác sĩ, tài khoản người dùng và nhật ký hệ thống. Việc tạo, sửa hoặc phân quyền tài khoản đều được kiểm soát tập trung bởi vai trò này.

Về xác thực, hệ thống sử dụng JWT (JSON Web Token) cho các API backend, kết hợp refresh token để duy trì phiên đăng nhập an toàn. Mật khẩu người dùng được băm

bằng các thuật toán mạnh như bcrypt hoặc argon2, giúp bảo vệ thông tin xác thực ngay cả khi dữ liệu bị rò rỉ. Cơ chế phân quyền chặt chẽ này đảm bảo mỗi người dùng chỉ thực hiện các thao tác phù hợp với vai trò được cấp.

3.5.2. Bảo mật dữ liệu bệnh nhân

Do dữ liệu y tế có mức độ nhạy cảm cao, module bốc số và quản lý hàng đợi được thiết kế tuân thủ ba nguyên tắc cốt lõi: tính bí mật (Confidentiality), toàn vẹn (Integrity) và sẵn sàng (Availability).

Toàn bộ giao tiếp giữa kiosk, server, dashboard và các phòng khám đều sử dụng giao thức HTTPS với TLS 1.2/1.3. Các kết nối thời gian thực như WebSocket cũng được mã hóa qua TLS nhằm ngăn chặn nguy cơ nghe lén hoặc đánh cắp token xác thực.

Dữ liệu lưu trữ được bảo vệ ở nhiều mức. Các thông tin nhận dạng như patient_id, embedding khuôn mặt và chỉ mục FAISS được lưu tách biệt trong mô-đun Face ID và không cho phép truy cập trực tiếp từ giao diện web. Các tệp nhạy cảm được bảo vệ bằng phân quyền hệ điều hành hoặc mã hóa ổ đĩa, kèm theo cơ chế sao lưu định kỳ và đối chiếu checksum để phát hiện lỗi hoặc chỉnh sửa trái phép.

Hệ thống cũng giới hạn mức độ hiển thị dữ liệu theo từng ngữ cảnh sử dụng. Tại dashboard công khai, chỉ hiển thị số thứ tự và phòng khám mà không hiển thị tên bệnh nhân. Giao diện phòng khám chỉ cho phép bác sĩ xem dữ liệu liên quan đến phòng của mình. Ảnh khuôn mặt chỉ xuất hiện trong quá trình nhận diện và không được hiển thị trên màn hình gọi số hay dashboard.

Mọi API có khả năng thay đổi trạng thái hàng đợi hoặc phiên khám đều yêu cầu xác thực token và kiểm tra vai trò. Điều này giúp ngăn chặn các hành vi gọi API trái phép hoặc tạo yêu cầu ảo nhằm chiếm lượt khám. Ngoài ra, hệ thống ghi log toàn bộ các thao tác quan trọng và phát cảnh báo khi phát hiện hành vi bất thường như đăng nhập sai nhiều lần hoặc truy cập API không đúng vai trò.

3.5.3. Tối ưu hiệu năng và khả năng chịu tải

Module bốc số và quản lý hàng đợi được thiết kế để đáp ứng môi trường bệnh viện có lưu lượng truy cập lớn, đặc biệt vào giờ cao điểm. Do đó, hệ thống phải đảm bảo khả năng xử lý nhanh, ổn định và không xảy ra nghẽn.

Về dữ liệu hàng đợi, các bảng wait_queue và real_time được thiết kế theo dạng bảng phẳng (wide table), cho phép truy cập nhanh theo cột phòng khám. Thuật toán phân bổ slot dựa trên thứ tự wait_index, giúp giảm độ phức tạp xử lý. Cơ chế cache, chẳng hạn như lưu số thứ tự cuối cùng theo từng dịch vụ, được sử dụng để hạn chế truy vấn lặp lại vào cơ sở dữ liệu.

Đối với giao diện điều phối, hệ thống sử dụng WebSocket thời gian thực thay cho cơ chế polling truyền thống. Mỗi khi có thay đổi trong hàng đợi thực tế, server chủ động đẩy bản cập nhật đến tất cả client, giúp giảm tải backend và đảm bảo độ trễ cập nhật ở mức rất thấp.

Để xử lý đồng thời và tránh xung đột, hệ thống áp dụng các khóa logic ở tầng ứng dụng khi nhiều kiosk cùng gửi yêu cầu bốc số. Cơ chế này đảm bảo không xảy ra trùng số thứ tự và tránh xung đột khi nhiều bác sĩ thao tác trên cùng một phòng khám.

Module Face ID được tối ưu bằng cách xây dựng sẵn chỉ mục FAISS cho toàn bộ embedding, cho phép truy vấn nhận diện trong thời gian rất ngắn. Ảnh đầu vào được nén và chuẩn hóa trước khi đưa vào pipeline nhằm giảm chi phí tính toán.

Về khả năng mở rộng, hệ thống hỗ trợ cả mở rộng chiều ngang (nhiều kiosk, nhiều worker backend, message broker cho realtime) và mở rộng chiều dọc (nâng cấp CPU, GPU, RAM). Đồng thời, các cơ chế dự phòng như sao lưu nóng cơ sở dữ liệu và tự khởi động lại worker khi gặp lỗi được triển khai để đảm bảo tính sẵn sàng cao. Trong trường hợp module Face ID tạm thời ngừng hoạt động, hệ thống vẫn cho phép bốc số thủ công, đảm bảo quy trình tiếp nhận bệnh nhân không bị gián đoạn.

CHƯƠNG 4: THỬ NGHIỆM VÀ ĐÁNH GIÁ API

4.1. Kịch bản và môi trường thử nghiệm

Môi trường thử nghiệm của hệ thống được triển khai trên một máy tính cá nhân đóng vai trò server thử nghiệm, nhằm mô phỏng điều kiện vận hành thực tế tại các phòng khám và bệnh viện quy mô vừa. Thiết bị sử dụng có cấu hình gồm CPU Intel® Core™ i7-11370H (4 nhân, 8 luồng, xung nhịp 3.30–4.8 GHz), RAM 16 GB DDR4, GPU NVIDIA GeForce RTX 3060 Laptop (6 GB VRAM), ổ cứng SSD NVMe 512 GB, chạy hệ điều hành Windows 11 64-bit và kết nối mạng Wi-Fi băng tần 5 GHz trong mạng nội bộ.

Hệ thống kiosk và camera được mô phỏng bằng camera USB độ phân giải 1080p kết hợp với màn hình cảm ứng 15.6 inch, kết nối qua LAN hoặc Wi-Fi tùy theo từng phiên thử nghiệm. Cấu hình phần cứng này được lựa chọn do đáp ứng tốt yêu cầu vận hành backend FastAPI kết hợp FAISS và WebSocket. Đồng thời, GPU RTX 3060 cho phép đánh giá khả năng tăng tốc của ONNX Runtime đối với mô-đun nhận diện khuôn mặt. Đây cũng là cấu hình phổ biến trong thực tế, giúp kết quả thử nghiệm có tính đại diện cao.

Về môi trường phần mềm, backend của hệ thống được xây dựng trên FastAPI kết hợp Uvicorn. Mô-đun định danh khuôn mặt sử dụng thư viện InsightFace với mô hình ArcFace (buffalo_l), được thử nghiệm ở hai chế độ CPUExecutionProvider và CUDAExecutionProvider nhằm so sánh hiệu năng xử lý. Thư viện FAISS được sử dụng cho bài toán tìm kiếm vector với chỉ mục FlatL2 trên embedding 512 chiều. Cơ sở dữ liệu MySQL 8.0 đảm nhiệm lưu trữ dữ liệu hàng đợi, thông tin bệnh nhân và nhật ký hệ thống. Giao diện người dùng được phát triển bằng Vue.js 3 kết hợp Vite và WebSocket, hỗ trợ cập nhật dữ liệu theo thời gian thực.

Các kịch bản thử nghiệm được thiết kế dựa trên điều kiện vận hành thực tế. Đối với mô-đun nhận diện khuôn mặt, hệ thống sử dụng tập dữ liệu gồm 300 ảnh của 100 bệnh nhân, mỗi bệnh nhân có ba góc chụp: chính diện, nghiêng trái và nghiêng phải. Tốc độ và độ chính xác nhận diện được đo trong các điều kiện đủ sáng, ánh sáng yếu và khuôn mặt bị che một phần, trên cả CPU và GPU.

Kịch bản bốc số và điều phối mô phỏng từ 50 đến 200 bệnh nhân mỗi giờ, mỗi bệnh nhân đăng ký từ một đến ba dịch vụ khám, với 10 phòng khám và 8 bác sĩ hoạt động đồng thời. Các tiêu chí đánh giá bao gồm tính công bằng theo nguyên tắc "đến trước – phục vụ trước", không xảy ra trùng lặp bệnh nhân trong hàng đợi tạm (wait_queue) và thời gian chuyển bệnh nhân sang hàng đợi thời gian thực (real_time).

Ngoài ra, dashboard thời gian thực được kiểm thử với 10 client hoạt động đồng thời, trong đó bác sĩ thực hiện thao tác gọi bệnh nhân liên tục mỗi 10 giây để đo độ trễ cập nhật của WebSocket. Hệ thống cũng được kiểm thử chịu tải khi kiosk gửi 20 yêu

cầu mỗi giây trong vòng 60 giây và xử lý liên tục từ 500 đến 2000 truy vấn Face ID. Kết quả benchmark cho thấy thời gian xử lý trung bình của ONNX Runtime đạt khoảng 50–90 ms/ảnh trên CPU và 8–15 ms/ảnh trên GPU RTX 3060.

Trong suốt quá trình thử nghiệm, các chỉ số như mức sử dụng CPU, RAM, GPU, thời gian xử lý trung bình của từng API, tốc độ truy xuất MySQL và độ trễ cập nhật dashboard được thu thập nhằm đánh giá độ chính xác, tính ổn định và khả năng mở rộng của hệ thống.

4.2. Kết quả thử nghiệm chức năng

4.2.1. Kết quả nhận diện khuôn mặt

Bảng 4.1. Kết quả nhận diện gương mặt

Trường hợp thử nghiệm	Tỉ lệ nhận diện đúng	Thời gian trung bình
Ảnh chính diện (đủ sáng)	Khoảng 99%	55–70 ms
Ảnh góc nghiêng ≤ 20°	95.1	600–88 ms
Ảnh nghiêng 20–30°	89.4%	60–88 ms
Ánh sáng yếu	87.2%	70–100 ms

Nhận xét:

Kết quả cho thấy mô-đun ArcFace kết hợp FAISS hoạt động ổn định với độ chính xác cao trong môi trường phòng khám. Trong điều kiện ánh sáng yếu hoặc góc nghiêng lớn, độ chính xác có giảm nhưng vẫn nằm trong ngưỡng chấp nhận được. Thời gian nhận diện trung bình dưới 100 ms đáp ứng tốt yêu cầu vận hành của kiosk tự phục vụ, mang lại trải nghiệm gần như tức thì cho người dùng.

4.2.2. Kết quả bốc số và điều phối bệnh nhân

Kết quả thử nghiệm cho thấy hệ thống bốc số và điều phối bệnh nhân hoạt động chính xác và ổn định theo đúng mục tiêu thiết kế. Toàn bộ bệnh nhân được gán đúng dịch vụ, đúng chuyên khoa và đúng phòng khám tương ứng dựa trên bảng ánh xạ chuyên khoa của hệ thống, đạt tỷ lệ chính xác 100%. Lỗi trước đây, trong đó bệnh nhân bị dồn vào cùng một phòng khám, đã được khắc phục hoàn toàn sau khi thuật toán điều phối được cải tiến.

Thuật toán quản lý hàng đợi đảm bảo không xảy ra hiện tượng trùng lặp bệnh nhân trong cùng một hàng đợi tạm. Qua kiểm chứng với 10.000 dòng dữ liệu mô phỏng, không ghi nhận trường hợp một mã bệnh nhân (patient_id) xuất hiện ở nhiều vị trí trong cùng một dòng dữ liệu, xác nhận cơ chế phân bố hàng đợi hoạt động chính xác.

Cơ chế điều phối hai lớp vận hành đúng theo thiết kế: lớp thứ nhất sắp xếp bệnh nhân vào hàng đợi tạm theo dịch vụ đăng ký, lớp thứ hai tự động chuyển bệnh nhân sang hàng đợi thời gian thực khi phòng khám sẵn sàng tiếp nhận. Khi bác sĩ gọi bệnh nhân, các thành phần liên quan như hàng đợi thời gian thực, nhật ký tiếp nhận và dashboard điều phối được cập nhật gần như tức thời với độ trễ dưới 200 ms.

Các thử nghiệm về tính công bằng cho thấy hệ thống luôn duy trì đúng nguyên tắc "đến trước – phục vụ trước". Với kịch bản 100 bệnh nhân đăng ký liên tục, thứ tự vào khám được bảo toàn mà không xảy ra sai lệch.

4.3. Đánh giá hiệu năng các API

4.3.1. Thời gian xử lý trung bình một lượt

Bảng 4.2. Thời gian xử lý trung bình một lượt

Tác vụ	Thời gian thực thi
Nhận diện Face ID	55–90 ms
Tra cứu dịch vụ + sinh số thứ tự	5–10 ms
Gán vào wait_queue	2–5 ms
Rót sang real_time	5–12 ms
Cập nhật dashboard real-time	50–150 ms

Tổng thời gian từ lúc bệnh nhân đứng trước kiosk đến khi nhận được số khám nhỏ hơn 0.3 giây, nhanh hơn đáng kể so với quy trình tiếp nhận truyền thống.

4.3.2. Khả năng hoạt động khi số lượng bệnh nhân tăng

Hệ thống duy trì hoạt động ổn định khi số lượng bệnh nhân tăng ở nhiều mức khác nhau. Trong kịch bản từ 30 đến 50 bệnh nhân mỗi phút, không ghi nhận độ trễ đáng kể, mức sử dụng CPU của backend dao động trong khoảng 8–12%. Khi tải tăng lên khoảng 100 bệnh nhân mỗi phút, hệ thống vẫn vận hành ổn định, kết nối WebSocket duy trì tốc độ cập nhật 10–15 lần mỗi giây, trong khi thời gian xử lý mỗi lượt chỉ tăng nhẹ khoảng 3–5 ms.

Với kịch bản tải cao gồm khoảng 500 truy vấn nhận diện liên tục, mô-đun tìm kiếm vector dựa trên FAISS vẫn giữ được hiệu năng ổn định. Mặc dù mức sử dụng CPU tăng, hệ thống không xuất hiện tình trạng quá tải hay gián đoạn dịch vụ. Điều này cho thấy hệ thống có khả năng đáp ứng nhu cầu của các phòng khám và bệnh viện quy mô vừa trong điều kiện thực tế.

4.4. Đánh giá hiệu quả ứng dụng trong bối cảnh chuyển đổi số bệnh viện

4.4.1. So sánh quy trình cũ và quy trình có hệ thống đề xuất

Bảng 4.3. So sánh quy trình cũ và mới

Tiêu chí	Quy trình cũ	Hệ thống đề xuất

Thời gian tiếp nhận	2-5 phút/người	30-90 giây/người
Nhận dạng bệnh nhân	Dựa vào giấy tờ, dễ sai	Face ID tự động
Sai lệch thông tin	Có thể xảy ra.	Gần 0%
Quản lý hàng đợi	Thủ công dễ trùng	Điều phối 2 lớp, tự động
Gọi bệnh nhân	Thủ công, phải hét số	Tự động hiển thị trên màn hình
Theo dõi trạng thái	Không đồng bộ	Dashboard realtime
Báo cáo thống kê	Làm tay, mất thời gian	Ghi log tự động

Quy trình mới giúp rút ngắn 70–90% thời gian tiếp nhận, giảm đáng kể sai sót do nhập liệu thủ công và nâng cao tính minh bạch, công bằng trong xếp hàng.

4.4.2. Lợi ích đối với bệnh nhân, nhân viên tiếp nhận, bác sĩ

Đối với bệnh nhân, hệ thống giúp đơn giản hóa quy trình tiếp nhận, giảm phụ thuộc vào giấy tờ và hạn chế tình trạng chen lấn tại quầy. Việc hiển thị rõ ràng số thứ tự và trạng thái khám giúp giảm thời gian chờ đợi và cải thiện trải nghiệm khám chữa bệnh.

Đối với nhân viên tiếp nhận, hệ thống giảm đáng kể khối lượng nhập liệu thủ công, hạn chế sai sót và hỗ trợ quản lý nhiều dịch vụ khám đồng thời. Thông tin trực quan về tình trạng tải của từng phòng khám giúp việc điều phối bệnh nhân hiệu quả hơn.

Đối với bác sĩ, hệ thống cung cấp danh sách bệnh nhân rõ ràng, cập nhật theo thời gian thực, hỗ trợ gọi bệnh nhân nhanh chóng và chính xác. Nhật ký lượt khám giúp theo dõi và kiểm soát lưu lượng bệnh nhân trong suốt ca làm việc.

4.5. Nhận xét chung về hệ thống

Đối với bệnh nhân, hệ thống giúp đơn giản hóa quy trình tiếp nhận, giảm phụ thuộc vào giấy tờ và hạn chế tình trạng chen lấn tại quầy. Việc hiển thị rõ ràng số thứ tự và trạng thái khám giúp giảm thời gian chờ đợi và cải thiện trải nghiệm khám chữa bệnh.

Đối với nhân viên tiếp nhận, hệ thống giảm đáng kể khối lượng nhập liệu thủ công, hạn chế sai sót và hỗ trợ quản lý nhiều dịch vụ khám đồng thời. Thông tin trực quan về tình trạng tải của từng phòng khám giúp việc điều phối bệnh nhân hiệu quả hơn.

Đối với bác sĩ, hệ thống cung cấp danh sách bệnh nhân rõ ràng, cập nhật theo thời gian thực, hỗ trợ gọi bệnh nhân nhanh chóng và chính xác. Nhật ký lượt khám giúp theo dõi và kiểm soát lưu lượng bệnh nhân trong suốt ca làm việc.

III. PHẦN KẾT LUẬN

1. Kết luận chung

Luận văn đã nghiên cứu, thiết kế và hiện thực thành công một tập các API và kiến trúc hệ thống phục vụ bài toán định danh bệnh nhân bằng công nghệ nhận diện khuôn mặt, kết hợp với mô-đun bốc số và điều phối hàng đợi khám bệnh tự động. Các API được xây dựng nhằm hỗ trợ quy trình tiếp nhận bệnh nhân theo hướng giảm thao tác thủ công, tăng mức độ tự động hóa và cải thiện khả năng điều phối trong môi trường bệnh viện.

Kết quả đạt được cho thấy hệ thống API đề xuất có khả năng nhận diện bệnh nhân với độ chính xác tương đối cao trong điều kiện tiêu chuẩn, hỗ trợ đăng ký và bốc số cho nhiều dịch vụ khám khác nhau, đồng thời đảm bảo tính nhất quán và công bằng trong thứ tự khám. Mô hình điều phối hàng đợi hai lớp, bao gồm hàng chờ tạm thời (wait queue) và hàng chờ thời gian thực trước phòng khám (real-time queue), giúp hạn chế trùng lặp lượt khám và giảm nguy cơ tắc nghẽn trong quá trình vận hành.

Bên cạnh đó, việc xây dựng dashboard điều phối và cơ chế cập nhật thời gian thực thông qua WebSocket/SSE giúp nhân viên y tế có cái nhìn tổng quan về tình trạng khám bệnh, số lượng bệnh nhân đang chờ và tiến độ khám tại từng phòng. Qua đó, hệ thống API đã chứng minh được tính khả thi về mặt kỹ thuật và cho thấy tiềm năng ứng dụng trong thực tế khi được tích hợp vào hạ tầng công nghệ thông tin của bệnh viện.

Tuy nhiên, trong phạm vi của luận văn, hệ thống mới dừng lại ở mức nguyên mẫu (prototype) và tập trung chủ yếu vào thiết kế kiến trúc cũng như hiện thực các API cốt lõi. Việc triển khai đồng bộ trên quy mô lớn và tích hợp trực tiếp với hệ thống HIS/EHR thực tế chưa được thực hiện, nhưng kết quả đạt được là nền tảng quan trọng cho các bước phát triển tiếp theo.

2. Các đóng góp chính của đề tài

Luận văn đã mang lại một số đóng góp nổi bật cả về mặt học thuật lẫn tính ứng dụng thực tiễn:

- Đề xuất và thiết kế kiến trúc hệ thống dựa trên API cho bài toán định danh bệnh nhân bằng Face ID kết hợp quản lý hàng đợi khám bệnh, phù hợp với xu hướng chuyển đổi số và tích hợp hệ thống trong lĩnh vực y tế.
- Hiện thực mô-đun Face ID dưới dạng dịch vụ độc lập, sử dụng embedding khuôn mặt và cơ sở dữ liệu vector kết hợp FAISS, giúp tăng tốc độ so khớp và đảm bảo khả năng mở rộng.
- Xây dựng và hiện thực mô hình bốc số và điều phối hai lớp, góp phần giải quyết các hạn chế thường gặp của hệ thống xếp hàng truyền thống như trùng

lặp bệnh nhân, nghẽn hàng đợi hoặc phân bổ không hợp lý giữa các phòng khám.

- Thiết kế cấu trúc dữ liệu và API phục vụ quản lý hàng đợi đa dịch vụ, đa phòng khám, đồng thời phát triển dashboard điều phối hỗ trợ theo dõi trạng thái khám bệnh theo thời gian thực.
- Đánh giá sơ bộ khả năng mở rộng và hiệu năng của hệ thống API, cho thấy nền tảng này có thể tiếp tục được phát triển và hoàn thiện để tiến tới triển khai thực tế.

3. Hạn chế của hệ thống

Bên cạnh các kết quả đạt được, hệ thống vẫn còn tồn tại một số hạn chế. Trước hết, độ chính xác của mô-đun nhận diện khuôn mặt có thể suy giảm trong các điều kiện không thuận lợi như ánh sáng yếu, góc mặt lớn hoặc khuôn mặt bị che khuất. Ngoài ra, các mô-đun Face ID, kiosk và dashboard hiện mới được kết nối thông qua các API ở mức cơ bản, chưa được chuẩn hóa đầy đủ theo các chuẩn công nghiệp cho hệ thống y tế quy mô lớn.

Hệ thống cũng chưa được tích hợp trực tiếp với HIS/EHR của một bệnh viện cụ thể, do đó chưa thể đánh giá toàn diện khả năng tương thích và vận hành trong môi trường thực tế. Thuật toán điều phối hàng đợi hiện tại chưa xét đến các trường hợp ưu tiên đặc thù như cấp cứu, tái khám theo lịch hẹn hoặc các nhóm bệnh nhân ưu tiên. Bên cạnh đó, mô-đun Face ID chưa được tích hợp các kỹ thuật chống giả mạo ảnh hoặc video (anti-spoofing), và hệ thống chưa có các công cụ phân tích dữ liệu chuyên sâu phục vụ quản trị bệnh viện.

4. Hướng phát triển

Từ nền tảng API và kiến trúc đã xây dựng, hệ thống có nhiều hướng phát triển trong tương lai. Trước hết, mô-đun Face ID có thể được tối ưu hiệu năng thông qua các kỹ thuật như quantization, sử dụng ONNX Runtime, TensorRT hoặc khai thác GPU để tăng tốc xử lý. Việc huấn luyện mô hình nhận diện khuôn mặt phù hợp hơn với dữ liệu khuôn mặt người Việt, đồng thời tích hợp mô-đun chống giả mạo ảnh và video, là những hướng cải tiến quan trọng nhằm nâng cao độ tin cậy của hệ thống.

Bên cạnh đó, hệ thống có thể được mở rộng và tích hợp sâu hơn với HIS/EHR, hướng tới xây dựng một nền tảng tiếp nhận bệnh nhân thống nhất, cho phép đồng bộ thông tin phòng khám, dịch vụ và lịch làm việc của bác sĩ. Việc phát triển các ứng dụng web hoặc di động cho phép bệnh nhân theo dõi số thứ tự, thời gian chờ và lịch khám từ xa cũng là một hướng đi tiềm năng.

Ngoài ra, dữ liệu thu thập từ hệ thống bốc số và điều phối có thể được khai thác để xây dựng các mô-đun phân tích và dự báo, chẳng hạn như dự đoán thời gian chờ, phân

tích lưu lượng bệnh nhân theo khung giờ và chuyên khoa, từ đó hỗ trợ tối ưu hóa việc bố trí nhân lực và phòng khám. Trong dài hạn, việc kết hợp thêm các thiết bị phần cứng như vòng tay thông minh hoặc kiosk tự phục vụ nâng cao có thể góp phần hình thành một hệ sinh thái tiếp nhận và quản lý bệnh nhân thông minh, đồng bộ và hiệu quả hơn.

TÀI LIỆU THAM KHẢO

[1] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, 2015.

[2] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4690–4699, 2019.

[3] X. Wu, R. He, Z. Sun, and T. Tan, "A Light CNN for Deep Face Representation with Noisy Labels," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2884–2896, 2018.

[4] S. Chen, Y. Liu, X. Gao, and Z. Han, "MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices," *arXiv preprint arXiv:1804.07573*, 2018.

[5] InsightFace Contributors, "InsightFace: 2D and 3D Face Analysis Project," GitHub Repository. Available: https://github.com/deepinsight/insightface (accessed 2025).

[6] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5203–5212, 2020.

[7] Ultralytics, "YOLOv8: State-of-the-Art Object Detection," GitHub Repository. Available: https://github.com/ultralytics/ultralytics (accessed 2025).

[8] J. Johnson, M. Douze, and H. Jégou, "Billion-Scale Similarity Search with FAISS," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.

[9] M. Douze, J. Johnson, and H. Jégou, "Indexing Millions of Images with Approximate Nearest Neighbor Search," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.

[10] M. Guo and M. Zipkin, "Analysis and Optimization of Patient Flow," *Operations Research*, vol. 66, no. 4, pp. 811–828, 2018.

[11] Y. Wang, "Design of an Intelligent Queue Management System Based on IoT and Cloud Computing," *IEEE Access*, vol. 9, pp. 12345–12356, 2021.

[12] N. Castañeda and A. M. Ross, "Queueing Models for Outpatient Clinics: A Review," *Health Systems*, vol. 5, no. 1, pp. 1–14, 2016.

[13] Bộ Y tế Việt Nam, *Chiến lược chuyển đổi số ngành Y tế giai đoạn 2025–2030*, Quyết định số 3516/QĐ-BYT, Hà Nội, 2025.

[14] Thủ tướng Chính phủ, *Chương trình chuyển đổi số quốc gia đến năm 2025, định hướng đến năm 2030*, Quyết định số 749/QĐ-TTg, Hà Nội, 2020.

[15] Bộ Y tế Việt Nam, *Thông tư số 13/2025/TT-BYT về triển khai bệnh án điện tử*, Hà Nội, 2025.

[16] World Health Organization, *Digital Health Systems and Interoperability*, WHO Technical Report, 2023.

[17] Healthcare Information and Management Systems Society (HIMSS), *Electronic Health Records and Healthcare Interoperability*, 2022.

[18] S. Kluyver et al., "FastAPI Documentation," Available: https://fastapi.tiangolo.com (accessed 2025).

[19] Vue.js Team, "Vue.js 3 Documentation," Available: https://vuejs.org (accessed 2025).

[20] Oracle Corporation, *MySQL 8.0 Reference Manual*, Available: https://dev.mysql.com/doc (accessed 2025).

[21] N. Jain, "Human Faces Dataset," *Kaggle Dataset*, 2021. Available: https://www.kaggle.com/datasets/niveditjain/human-faces-dataset

[22] S. Jain and A. Kumar, "Biometric Recognition in Healthcare: Applications and Challenges," *IEEE Access*, vol. 8, pp. 215123–215135, 2020.

[23] NYU Langone Health, "Biometric Check-in System for Patient Identification," Technical Overview, 2024.