HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Xuân Bách

HỆ THỐNG XÁC THỰC KHUÔN MẶT CHO ỨNG DỤNG DI ĐỘNG

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(*Theo định hướng ứng dụng*)

HÀ NỘI - NĂM 2025

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Xuân Bách

HỆ THỐNG XÁC THỰC KHUÔN MẶT CHO ỨNG DỤNG DI ĐỘNG

Chuyên ngành: Hệ thống thông tin
Mã số: 8480104

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT
(*Theo định hướng ứng dụng*)

NGƯỜI HƯỚNG DẪN KHOA HỌC:
TS. NGUYỄN DUY PHƯƠNG

HÀ NỘI - NĂM 2025

LỜI CAM ĐOAN

Tôi xin cam đoan đã thực hiện kiểm tra mức độ tương đồng nội dung đề án qua phần mềm kiểm tra tài liệu một cách trung thực và đạt kết quả mức độ tương đồng 6 % toàn bộ nội dung đề án tốt nghiệp. Bản đề án tốt nghiệp kiểm tra qua phần mềm là bản cứng đề án tốt nghiệp đã nộp để bảo vệ trước hội đồng. Nếu sai tôi xin chịu các hình thức kỷ luật theo quy định hiện hành của Học viện.

Hà Nội..., ngày 04 tháng 08 năm 2025

HỌC VIÊN CAO HỌC

(Ký và ghi rõ họ tên)

Bách

Nguyễn Xuân Bách

MỤC LỤC

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

Chữ viết tắt	Tiếng Anh	Tiếng Việt
API	Application Programming Interface	Giao diện lập trình ứng dụng
CNN	Convolutional Neural Network	Mạng nơ-ron tích chập
MTCNN	Multi-task Cascaded Convolutional Neural Network	Mạng nơ-ron tích chập bậc thang đa nhiệm
IoT	Internet of Things	Internet vạn vật
SQL	Structured Query Language	Ngôn ngữ truy vấn có cấu trúc
ReLU	Rectified Linear Unit	Hàm tuyến tính có chỉnh
CLAHE	Contrast Limited Adaptive Histogram Equalization	Cân bằng histogram thích nghi có giới hạn độ tương phản
GEMM	General Matrix Multiplication	Phép nhân ma trận tổng quát
NMS	Non-Maximum Suppression	Loại bỏ cực đại không cần thiết

DANH SÁCH BẢNG

DANH SÁCH HÌNH VẼ

MỞ ĐẦU

1. Lý do lựa chọn đề tài

Khái quát vấn đề còn tồn tại:

- Thiếu khả năng xác minh người dùng chính chủ: Trong các ứng dụng có lưu trữ, sử dụng thông tin cá nhân của người dùng, như ứng dụng nhắn tin hoặc các ứng dụng nhà thông minh còn thiếu chức năng xác minh người dùng chính chủ, tức là xác minh người dùng đó là chủ của tài khoản đó. Nếu như một người khác không phải là chủ tài khoản mà lại biết được thông tin đăng nhập, hoặc bị mất điện thoại thì việc bị lộ dữ liệu cá nhân là không thể tránh khỏi.
- Ứng dụng vào các ứng dụng nhà thông minh: Việc sử dụng camera đã trở nên bình thường với các hộ gia đình. Việc tích hợp thêm xác thực khuôn mặt sẽ giúp tạo ra nhiều tự động thông minh, khiến cuộc sống thoải mái, dễ chịu hơn. Ví dụ như camera trước cổng nhà phát hiện chủ nhân đã về nhà, qua đó tự động mở khoá nhà, bật điện, bình nóng lạnh giúp chủ nhà.
- Trải nghiệm người dùng kém: Việc phải nhập mật khẩu nhiều lần trong app có thể khiến người dùng cảm thấy bất tiện, đặc biệt với người sử dụng thường xuyên. Một vấn đề khác nữa là người dùng có thể quên mật khẩu do đặt quá dài hoặc quá phức tạp, dẫn đến việc tài khoản có thể bị khoá hoặc yêu cầu khôi phục, thay đổi mật khẩu phức tạp.

Lý do lựa chọn đề tài:

- Trong bối cảnh chuyển đổi số và sự phát triển vượt bậc của công nghệ thông tin, nhu cầu đảm bảo bảo mật và xác thực danh tính cho các ứng dụng di động đang ngày càng trở nên cần thiết.
- Hơn nữa, xác thực khuôn mặt là một công nghệ tiên tiến giúp nhận diện và xác minh danh tính người dùng dựa trên đặc điểm sinh học của khuôn mặt. Công

nghệ này không chỉ tăng cường bảo mật mà còn mang đến trải nghiệm người dùng tiện lợi, nhanh chóng và không cần nhập liệu thủ công.

- Ngoài ra, với việc tích hợp thêm xác thực khuôn mặt cho các camera trong ứng dụng nhà thông minh góp phần tạo nên những tự động phù hợp với nhu cầu của các gia đình.
- Vì vậy, đề tài "Hệ thống xác thực khuôn mặt cho ứng dụng di động" được lựa chọn để cung cấp cho các nhà phát triển ứng dụng di động một cách để tích hợp chức năng xác thực khuôn mặt vào ứng dụng của mình, giúp đảm bảo quyền riêng tư cho người dùng, đồng thời giúp cuộc sống con người dễ dàng hơn.

2. Mục đích, ý nghĩa của đề tài

Mục tiêu của đề án

- Tạo ra được hệ thống xác thực khuôn mặt hiệu quả, tích hợp nhanh chóng và dễ dàng với các ứng dụng di động
- Tích hợp được các công cụ nhận diện, xác thực khuôn mặt vào hệ thống
- Giảm thời gian thực thi và tăng độ chính xác trong việc xác thực khuôn mặt

Các kết quả cần đạt được(về mặt lý luận và thực tiễn)

Về mặt lý luận

- Hiểu rõ hơn về các mô hình, phương pháp nhận diện, xác thực khuôn mặt, từ đó có thể cải tiến thuật toán, giúp rút ngắn thời gian thực thi hoặc tăng độ chính xác
- Hiểu rõ hơn về cách vận hành hệ thống

Về mặt thực tiễn

- Tạo ra được một hệ thống xác thực khuôn mặt hiệu quả
- Hệ thống có thể hoạt động tốt trên các loại smart phone khác nhau.
- Đánh giá, cải tiến các mô hình đã sử dụng

Ý nghĩa của đề tài: Nghiên cứu này có thể đóng góp vào việc phát triển các hệ thống tích hợp chức năng xác thực khuôn mặt, giúp các ứng dụng di động có thêm lớp bảo mật. Nó giúp thông tin của người dùng được bảo mật và tránh rò rỉ thông tin.

Tính cấp thiết, khả thi của đề tài:

- Tính cấp thiết: Các ứng dụng di động sử dụng dữ liệu của người dùng ngày càng nhiều. Để đảm bảo các dữ liệu trong ứng dụng không bị kẻ xấu đánh cắp và sử dụng cho mục đích cá nhân của chúng, cần có một hệ thống nhận diện, xác thực khuôn mặt dễ tích hợp, triển khai nhanh chóng .
- Tính khả thi: Hiện tại đã có nhiều mô hình phát hiện, nhận diện khuôn mặt được phát triển và đưa ra được kết quả với độ chính xác gần như tuyệt đối. Hơn nữa, phần lớn các ứng dụng nhắn tin, ứng dụng nhà thông minh hiện tại chưa tích hợp xác thực khuôn mặt, chưa có những ngữ cảnh, tự động theo sự kiện xác thực khuôn mặt.

3. Đối tượng, phương pháp nghiên cứu

Đối tượng nghiên cứu:

- Các mô hình, phương pháp nhận diện, xác thực khuôn mặt, cùng với các ứng dụng có tích hợp chúng trong các chức năng của ứng dụng.
- Các hệ thống tương tự đã có trong thực tế.
- Phạm vi nghiên cứu:
- Phạm vi không gian: Nghiên cứu sẽ được thực hiện trong các môi trường giả lập, hoặc có thể thực hiện trên một số ứng dụng di động ở nơi làm việc.
- Phạm vi thời gian: Nghiên cứu sẽ được thực hiện trong khoảng thời gian từ 4 đến 6 tháng, bao gồm việc xây dựng hệ thống và tích hợp các mô hình.

Phương pháp lựa chọn: Mô phỏng và thực nghiệm

- Xác định hệ thống ổn định và chạy bình thường
- Đánh giá độ chính xác của các mô hình, phương pháp nhận diện khuôn mặt sử dụng trong hệ thống.

Nghiên cứu tài liệu và khảo sát: Để xây dựng cơ sở lý thuyết, đồng thời có thể cải tiến để giảm thời gian xác thực hoặc tăng độ chính xác.

Các thí nghiệm và điều tra:

- Thí nghiệm nhận dạng khuôn mặt từ hình ảnh, video: Thực hiện chức năng nhận dạng khuôn mặt và đánh giá hiệu quả
- Thí nghiệm xác định khuôn mặt thật: Thực hiện chức năng xác định khuôn mặt dựa trên video, hình ảnh đã có nhận dạng khuôn mặt và đánh giá hiệu quả
- Thí nghiệm xác thực khuôn mặt: Thực hiện chức năng xác thực khuôn mặt bằng cách so sánh với dữ liệu đã có với dữ liệu nhận được từ video, hình ảnh gửi đến.
- Điều tra: Thu thập thông tin về các mô hình xác thực khuôn mặt, các hệ thống tương tự để hiểu rõ những vấn đề liên quan

Các công cụ, thiết bị được sử dụng:

- Thiết bị sử dụng: Đề án của tôi sử dụng điện thoại thông minh để thực hiện các chức năng nhận diện khuôn mặt cho ứng dụng.
- Xcode: IDE của Apple, cho phép các nhà phát triển xây dựng ứng dụng và triển khai lên điện thoại.
- VS Code: IDE giúp xây dựng hệ thống phát hiện, nhận diện khuôn mặt, nhận diện khuôn mặt giả mạo, và API.
- Ngôn ngữ lập trình: tôi sử dụng python cho những phần liên quan đến phát hiện, nhận diện khuôn mặt, nhận diện khuôn mặt giả mạo, xây dựng API. Ngoài ra, tôi còn sử dụng Swift để phát triển ứng dụng di động.
- Các công cụ triển khai server: ngrok, uvicorn.

4. Cấu trúc đề tài

Nội dung chính của đề án được chia thành 3 chương chính như sau:

- Chương 1: Cơ sở lý luận: Chương này tập trung vào việc phân tích các phương pháp, mô hình được sử dụng để xây dựng hệ thống như MTCNN, MobileNetV2, FaceNet
- Chương 2: Xây dựng hệ thống nhận diện khuôn mặt: Chương này tập trung vào việc thiết kế hệ thống, các module trong hệ thống.
- Chương 3: Triển khai và thử nghiệm: Chương này tập trung vào cách triển khai hệ thống lên ứng dụng nhà thông minh và đưa ra các kết quả thử nghiệm.
- Chương 4: Đánh giá và kết luận: Đánh giá được ưu điểm và nhược điểm còn tồn tại của hệ thống, đề xuất các nghiên cứu nâng cấp và cải thiện hệ thống.

CHƯƠNG 1: CƠ SỞ LÝ LUẬN

1.1. Phát hiện và căn chỉnh khuôn mặt bằng MTCNN

1.1.1. Định nghĩa

Phát hiện khuôn mặt là kỹ thuật thị giác máy tính dùng để tự động nhận diện, xác định vị trí của khuôn mặt trong hình ảnh, khung hình video bằng cách xác định toạ độ và kích thước chúng. Đây là bước đầu tiên trước các nhiệm vụ như nhận dạng khuôn mặt, phân tích biểu cảm. Thuật toán phát hiện khuôn mặt rất đa dạng, bao gồm sử dụng mạng neural, cascade classifier và các kỹ thuật dựa trên các đặc trưng để vượt qua các điều kiện khó khăn như ánh sáng, vật cản và chất lượng hình ảnh. Trong đề tài này, tôi sử dụng MTCNN – một kỹ thuật học sâu sử dụng mạng neural để phát hiện khuôn mặt phục vụ cho việc nhận diện khuôn mặt.

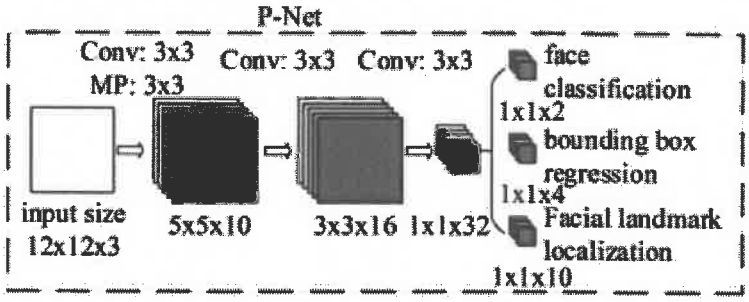
MTCNN (Mạng tích chập nhiều tác vụ) là một framework học sâu được thiết kế để đồng thời phát hiện khuôn mặt, căn chỉnh và định vị các điểm mốc của khuôn mặt. Không giống như các phương pháp truyền thống xử lý các tác vụ này riêng biệt, MTCNN tích hợp chúng bằng cách sử dụng các mạng CNN nối tiếp thống nhất (unified cascaded CNNs), cho phép xử lý hiệu quả và chính xác. Được giới thiệu vào năm 2016, thuật toán giải quyết các thách thức về kích thước khuôn mặt, hướng và điều kiện ánh sáng khác nhau bằng cách tận dụng phân tích hình ảnh ở nhiều tỷ lệ và tinh chỉnh dần dần. Khả năng phát hiện năm điểm mốc chính của khuôn mặt (mắt, mũi, khóe miệng) trong khi tạo các hộp giới hạn khiến nó trở nên không thể thiếu đối với các ứng dụng cần phân tích khuôn mặt chính xác.

1.1.2. Cấu tạo

MTCNN bao gồm 3 mạng neural tích chập (CNN) được sắp xếp thành từng tầng, mỗi mạng là một giai đoạn riêng biệt trong quá trình phát hiện khuôn mặt:

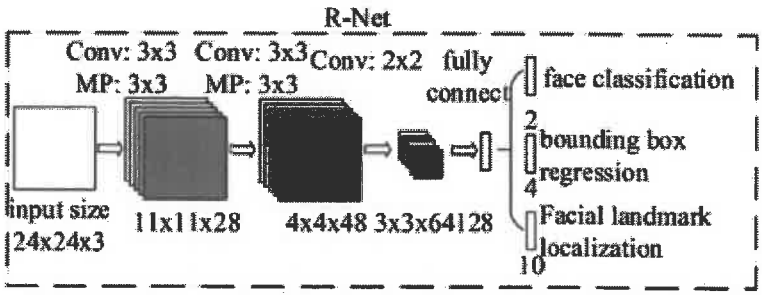
- P-Net (Proposal Network): Là giai đoạn đầu tiên, xử lý image piramid (là tập các ảnh với tỉ lệ khác nhau từ ảnh gốc, sắp xếp như một kim tự tháp) để tạo ra các

vùng ứng viên là khuôn mặt. Sử dụng cửa sổ trượt 12×12 pixel, P-Net áp dụng các bộ lọc tích chập và các lớp max pooling để chuyển đổi thành các đặc trưng sau đó chúng được phân tích để xác suất khuôn mặt và tọa độ hộp giới hạn. Các ứng viên có độ tin cậy thấp sẽ bị loại bỏ, giảm tải tính toán cho các giai đoạn tiếp theo [1].



Hình 1.1. Cấu tạo của P-Net [1]

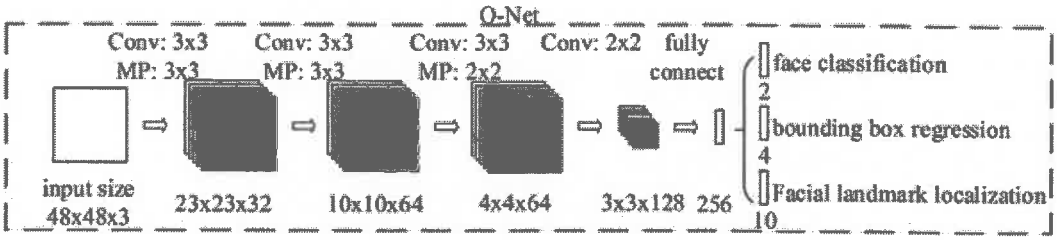
- R-Net (Refine Network): Là giai đoạn thứ hai, chấp nhận các vùng ứng viên từ P-Net, thay đổi kích thước của chúng thành 24×24 pixel và sử dụng mạng CNN sâu hơn để loại bỏ các kết quả sai. R-Net tinh chỉnh tọa độ hộp giới hạn bằng các lớp full connected, đảm bảo căn chỉnh với các đặc điểm khuôn mặt. Bước này cải thiện đáng kể độ chính xác phát hiện bằng cách lọc ra các vùng không phải khuôn mặt [1].



Hình 1.2. Cấu tạo R-Net [1]

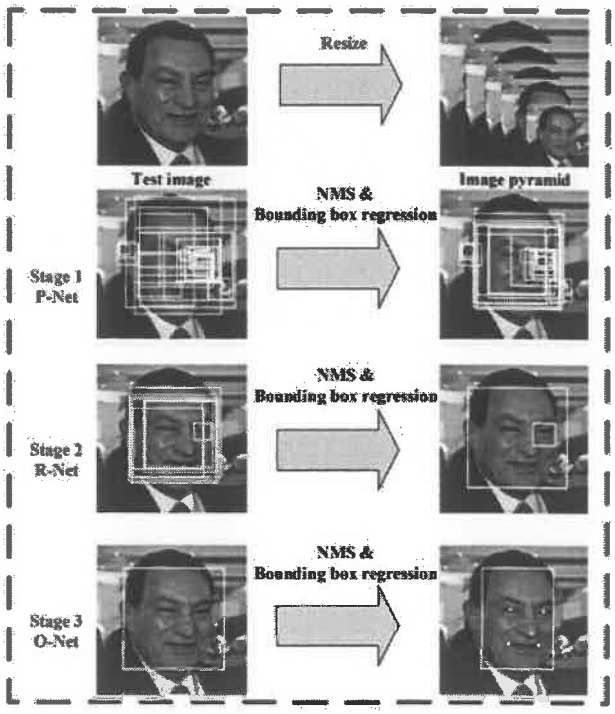
- O-Net (Output Network): Là giai đoạn cuối cùng, xử lý các vùng có kích thước 48×48 pixel qua một mạng CNN phức tạp hơn nhiều, đưa ra được chính xác các hộp giới hạn và toạ độ của 5 điểm mốc khuôn mặt. Cấu tạo của O-Net có thêm các lớp tích chập và lớp gộp tối đa, cho phép trích xuất đặc trưng chi tiết hơn và hoạt

đông tốt với các điều kiện xấu như bị che khuất một phần khuôn mặt hoặc ánh sáng không đồng đều [1].



Hình 1.3. Cấu tạo O-Net [1]

1.1.3. Nguyên lý hoạt động



Hình 1.4. Hoạt động của MTCNN [1]

MTCNN bắt đầu với việc xây dựng image pyramid, trong đó hình ảnh đầu vào được thay đổi kích thước theo nhiều tỉ lệ.

Ở P-Net, một cửa sổ có kích thước 12x12 pixel được quét qua toàn bộ các bức ảnh với bước nhảy 2 pixel và tạo ra những vùng ứng viên với độ tin cậy tương ứng. Những ứng viên có độ tin cậy trên ngưỡng sẽ tiếp tục qua bước điều chỉnh lại vị trí

hộp giới hạn bằng phương pháp hồi quy. Sau đó, sử dụng NMS để gộp các ứng viên trùng lặp. Kết quả đầu ra được đưa vào R-Net để tiếp tục tinh chỉnh.

Trong giai đoạn tiếp theo, R-Net tiếp tục loại bỏ các ứng viên không phải là khuôn mặt, điều chỉnh hộp giới hạn bằng phương pháp hồi quy và sử dụng NMS để gộp các ứng viên còn lại.

Trong giai đoạn cuối cùng, O-Net đảm nhận việc đưa ra vị trí 5 điểm mốc của khuôn mặt, cùng với đó là tiếp tục điều chỉnh hộp giới hạn bằng phương pháp hồi quy và sử dụng NMS để gộp các ứng viên còn lại.

Cách hoạt động của NMS:

- Sắp xếp tất cả các bounding box theo điểm số tin cậy giảm dần.
- Giữ lại bounding box có điểm số cao nhất.
- Loại bỏ tất cả các bounding box khác có IoU (Intersection over Union) với bounding box đã chọn cao hơn một ngưỡng nhất định.
- Lặp lại quá trình với các bounding box còn lại cho đến khi đã xét tất cả.

1.1.4. Ưu nhược điểm

Ưu điểm:

- MTCNN có độ chính xác phát hiện khuôn mặt đến 98.7% trên tập dữ liệu Winder Face [1].
- Khả năng phát hiện khuôn mặt cùng với căn chỉnh và xác định các điểm mốc giúp đơn giản hoá cho các tác vụ tiếp theo như nhận diện khuôn mặt, kiểm tra giả mạo khuôn mặt.
- Cấu trúc phân tầng của MTCNN giúp cho kết quả được chính xác mà vẫn cho phép nó hoạt động trong thời gian thực.

Nhược điểm:

- Độ phức tạp trong tính toán của MTCNN vẫn còn lớn đối với các ảnh có độ phân giải cao.

- Khó có thể phát hiện được các khuôn mặt khi ở các góc lạ (quay ngang hoàn toàn, cúi gằm hoặc ngửa quá mức) hoặc bị che khuất quá nhiều.

1.1.5. Ứng dụng thực tế

MTCNN được ứng dụng rộng rãi trong nhiều lĩnh vực nhờ khả năng phát hiện khuôn mặt chính xác và cung cấp thông tin chi tiết về điểm đặc trưng. Một số ứng dụng tiêu biểu bao gồm:

- Nhận diện khuôn mặt: MTCNN thường được sử dụng làm bước tiền xử lý trong các hệ thống nhận diện khuôn mặt, ví dụ như mở khóa điện thoại thông minh (Face ID trên iPhone) hoặc hệ thống điểm danh tự động.
- Phân tích biểu cảm khuôn mặt: MTCNN hỗ trợ xác định điểm đặc trưng để phân tích cảm xúc, được áp dụng trong các hệ thống quảng cáo cá nhân hóa hoặc nghiên cứu tâm lý học.
- An ninh và giám sát: MTCNN được tích hợp trong các hệ thống camera giám sát để phát hiện và theo dõi khuôn mặt trong đám đông, hỗ trợ nhận diện tội phạm hoặc tìm kiếm người mất tích.
- Y tế: Trong lĩnh vực y tế, MTCNN được sử dụng để phân tích các đặc điểm khuôn mặt nhằm phát hiện các dấu hiệu bệnh lý, như hội chứng Down hoặc các rối loạn di truyền.

1.2. Nhận diện khuôn mặt với FaceNet và MobileNet

1.2.1. Tổng quan về nhận diện khuôn mặt

Nhận diện khuôn mặt đã trở thành một trong những ứng dụng được nghiên cứu nhiều và sâu rộng, được triển khai rộng rãi trong lĩnh vực thị giác máy tính, gồm những hệ thống an ninh, giám sát điểm danh, kiểm soát truy cập. Đây là công nghệ sinh trắc học, sử dụng các phương pháp phân tích đặc điểm khuôn mặt để nhận dạng, xác nhận danh tính của người từ hình ảnh hoặc các khung hình video.

Quá trình này bao gồm việc trích xuất các đặc trưng khuôn mặt, so sánh chúng với cơ sở dữ liệu khuôn mặt có sẵn để xác định danh tính với độ chính xác cao. Bên cạnh đó, các phương pháp nhận dạng khuôn mặt vẫn còn phải giải quyết bài toán về điều kiện ánh sáng, sự lão hoá, sự khác biệt về dáng.

Trong những năm gần đây, các phương pháp học sâu đã được áp dụng vào hệ thống nhận dạng khuôn mặt, giúp cải thiện đáng kể độ chính xác và độ tin cậy so với các phương pháp truyền thống. Các hệ thống hiện đại có thể đạt được tỷ lệ nhận dạng ấn tượng ngay cả trong điều kiện thay đổi, nhờ việc tận dụng các kiến trúc mạng neural học sâu.

Trong quá trình triển khai của tôi, tôi đã sử dụng MobileNetV2 để trích xuất các vector face embedding và FaceNet để nhận diện khuôn mặt bằng cách so sánh độ tương đồng giữa các vector này. Mô hình này hoạt động tốt trên các thiết bị có hạn chế về tài nguyên, giúp tăng tốc xử lý, đồng thời giảm tính toán.

1.2.2. MobileNet

1.2.2.1. Tổng quan về MoblieNet

MobileNet là một kiến trúc mạng nơ-ron tích chập (CNN) hiệu quả, được thiết kế để có thể triển khai trên các thiết bị có tài nguyên hạn chế. Được phát triển bởi các nhà nghiên cứu của Google, MobileNet giải quyết thách thức này bằng cách sử dụng tích chập phân tách theo chiều sâu (depthwise separable convolution) để xây dựng một mạng nơ-ron tích chập nhẹ. Phương pháp phân tách này giúp giảm đáng kể chi phí tính toán và kích thước mô hình so với các phép tích chập tiêu chuẩn, cho phép thực hiện các tính toán thời gian thực trên thiết bị di động.

MobileNet mang lại sự cân bằng hiệu quả giữa độ trễ (latency) và độ chính xác (accuracy) thông qua các siêu tham số (hyperparameters) cho phép điều chỉnh mô hình phù hợp với các giới hạn tài nguyên cụ thể [7]. Nhờ các siêu tham số này, MobileNet có thể cân bằng giữa hiệu suất và chi phí tính toán, thích ứng được với

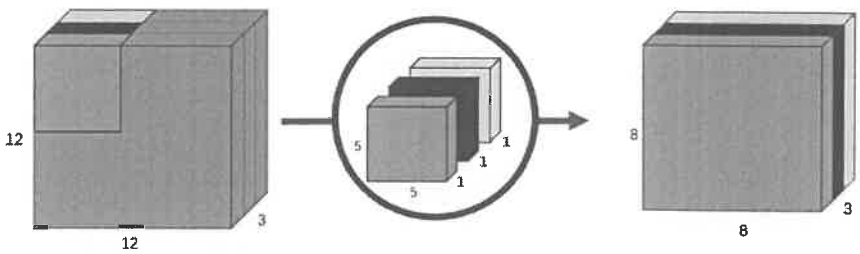
các phần cứng khác nhau, đồng thời duy trì độ chính xác chấp nhận được cho các ứng dụng nhận diện khuôn mặt.

Kiến trúc này thể hiện hiệu suất cạnh tranh trong nhiều nhiệm vụ thị giác máy tính như phân loại ảnh (image classification), phát hiện đối tượng (object detection), và nhận diện thuộc tính khuôn mặt (face attributes) [7], trong khi chỉ yêu cầu lượng tài nguyên tính toán ít hơn đáng kể.

1.2.2.2. Depthwise separable convolution

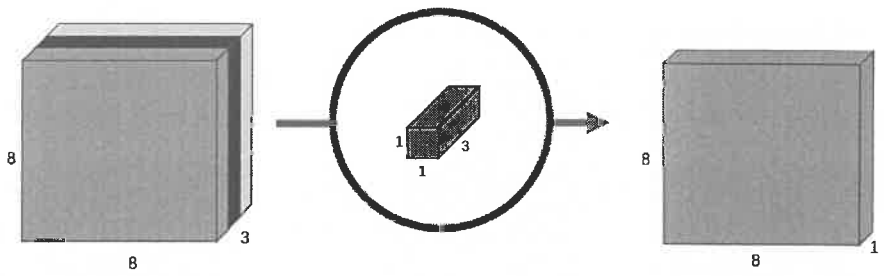
Mô hình MobileNet được xây dựng dựa trên depthwise separable convolution (mạng tích chập phân tách theo chiều sâu) bao gồm 2 bước:

- Depthwise convolution (tích chập theo chiều sâu): là một kiểu tích chập mà áp dụng từng bộ lọc tích chập cho mỗi kênh đầu vào. Với tích chập thông thường, bộ lọc có độ sâu bằng với số kênh đầu vào và tuỳ ý trộn các kênh để tạo từng phần tử trong đầu ra. Ngược lại, tích chập theo chiều sâu chia đầu vào thành các kênh, thực hiện tích chập với từng bộ lọc riêng biệt, sau đó xếp chồng các tích chập với nhau thành đầu ra [7].



Hình 1.5. Cấu tạo depthwise convolution

- Pointwise convolution (tích chập điểm): là kiểu tích chập 1x1: sử dụng kernel 1x1 để kết hợp các đặc trưng từ nhiều kênh của đầu ra của bước trước. Kernel có số kênh bằng với số kênh của ảnh đầu vào, mục đích để thu được 1 kênh của ảnh đầu ra. Số kênh đầu ra tương đương với số kernel cần có [7].



Hình 1.6. Cấu tạo pointwise convolution

Với depthwise separable convolution, MobileNet đã làm giảm được phần lớn chi phí tính toán so với tích chập tiêu chuẩn. Điều này làm nó trở nên phù hợp với các hệ thống có tài nguyên hạn chế.

Với tích chập tiêu chuẩn, chi phí tính toán sẽ là:

$$D_K . D_K . M . N . D_F . D_F$$

Trong đó:

- D_K là kích thước của kernel
- M là số kênh đầu vào,
- D_F là kích thước đặc trưng đầu ra
- N là số kênh đầu ra.

Còn với depthwise separable convolution, chi phí tính toán là tổng chi phí tính toán của depthwise convolution và pointwise convolution. Trong đó:

Chi phí của depthwise convolution là:

$$D_K . D_K . M . D_F . D_F$$

Chi phí của pointwise convolution là:

$$M . N . D_F . D_F$$

Chi phí của depthwise separable convolution là:

$$D_K . D_K . M . D_F . D_F + M . N . D_F . D_F$$

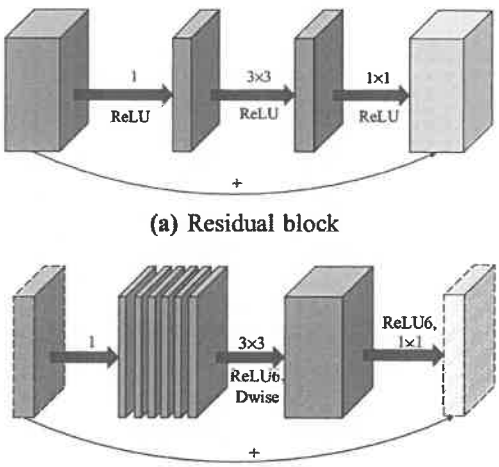
Từ đó ta thấy được tỉ lệ chi phí của depthwise separable convolution so với chi phí của tích chập tiêu chuẩn là:

$$\frac{D_K.D_K.M.D_F.D_F + M.N.D_F.D_F}{D_K.D_K.M.N.D_F.D_F} = \frac{1}{N} + \frac{1}{D_K^2}$$

Với kernel 3x3, tỉ lệ này là xấp xỉ 9 lần, giảm một lượng số lượng lớn tính toán cần thiết mà vẫn đảm bảo được số lượng đầu ra.

1.2.2.3. Cấu trúc inverted residual và linear bottlenecks

MobileNetV2 đã cách mạng hóa kiến trúc mạng neural cho thiết bị di động thông qua hai cải tiến chính: inverted residual blocks và linear bottlenecks. Khác với residual block truyền thống sử dụng mô hình "rộng-hẹp-rộng", inverted residual block đảo ngược thành "hẹp-rộng-hẹp" (hình 1.7). Cụ thể, mỗi block bắt đầu bằng lớp mở rộng (expansion layer) sử dụng phép tích chập pointwise 1×1 để tăng số kênh từ C_{in} lên $C_{exp} = t.C_{in}$ (với t là hệ số mở rộng). Tiếp theo, lớp depthwise convolution 3×3 thực hiện lọc đặc trưng không gian, và cuối cùng, lớp projection giảm số kênh về C_{out} thông qua pointwise convolution 1×1. Kết nối tắt (shortcut) chỉ được áp dụng khi stride = 1 và kích thước đầu vào/ra bằng nhau, giúp tối ưu hóa luồng gradient [8].



(a) Residual block

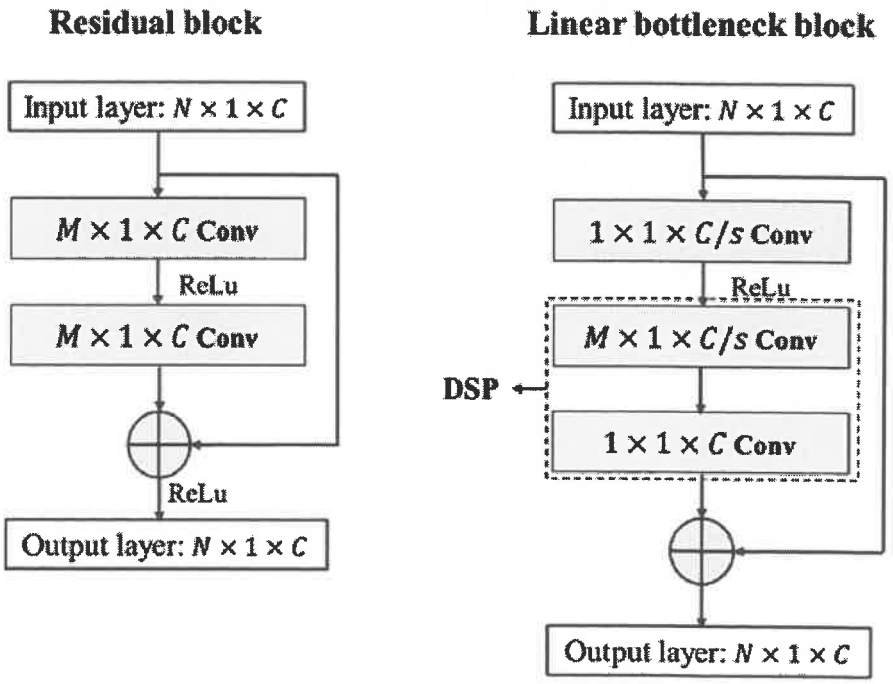
(b) Inverted residual block

Hình 1.7. Kiến trúc Inverted residual so với residual

Điểm đột phá thứ hai nằm ở việc loại bỏ hàm kích hoạt phi tuyến (ReLU) trong các lớp bottleneck mỏng. Nghiên cứu chỉ ra rằng ReLU gây mất thông tin trong không gian chiều thấp do tạo ra các manifold suy biến. Do đó, MobileNetV2 chỉ sử dụng hàm tuyến tính ở lớp projection, được biểu diễn qua công thức:

$$Y = F_{proj}$$

trong khi giữ ReLU6 cho các lớp mở rộng có chiều cao. Chiến lược này bảo toàn thông tin quan trọng và cải thiện độ ổn định của mô hình [8].



Hình 1.8. Kiến trúc Linear bottleneck block

Về hiệu quả tính toán, inverted residual block giảm đáng kể số tham số so với kiến trúc truyền thống. Tổng tham số cho một block được tính bằng:

$$\text{Params} = \text{Cin}.\text{Cexp} + 9.\text{Cexp} + \text{Cexp}.\text{Cout}$$

Với hệ số mở rộng t = 6, MobileNetV2 đạt độ chính xác 72.0% trên ImageNet chỉ với 300 triệu phép toán (MAdds), giảm 30% độ trễ so với phiên bản tiền nhiệm. Cấu trúc này còn linh hoạt khi điều chỉnh thông qua hệ số t, cho phép tối ưu cho các thiết bị có tài nguyên khác nhau. Trong thực tiễn, inverted residual và linear

bottlenecks đã mở đường cho các ứng dụng real-time như MobileDet (phát hiện vật thể) và Mobile DeepLabv3 (phân đoạn ngữ nghĩa), chứng minh khả năng cân bằng giữa độ chính xác và hiệu suất – yếu tố then chốt cho triển khai AI trên thiết bị edge.

1.2.2.4. Cấu tạo của MobileNetV2

Kiến trúc này phát triển dựa trên nền tảng của MobileNetV1, đồng thời mang đến những cải tiến mang tính đột phá về mặt cấu trúc, giúp cải thiện hiệu suất và độ hiệu quả tính toán một cách đáng kể. Điểm nổi bật nhất của MobileNetV2 chính là cấu trúc inverted residual kết hợp với linear bottleneck, cho phép mạng trích xuất đặc trưng mạnh mẽ nhưng vẫn giữ mức tiêu tốn tài nguyên thấp.

Dưới đây là bảng mô tả cấu tạo mô hình MobileNetV2. Trong đó:

- Input là Kích thước đầu vào của đặc trưng tại từng tầng
- Operator là loại tích chập hoặc khối được sử dụng
- t là hệ số mở rộng
- c là số lượng kênh đầu ra
- n là số lần lặp lại của loại tích chập hoặc khối
- s là số bước trong phép tích chập

Bảng 1.1. Mô hình MobileNet V2 [8]

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	160	3	1

$14^2 \times 64$	bottleneck	6	160	1	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1×1	-	1280	1	1
$7^2 \times 1280$	avgpool 7×7	-	-	-	-
$1 \times 1 \times 1280$	conv2d 1×1	-	k	-	-

Mạng được thiết kế theo hướng giảm dần độ phân giải và tăng dần số kênh:

- Input: ảnh RGB 224×224×3
- Conv đầu tiên: 32 kênh, stride 2
- Các bottleneck layer: 16 → 24 → 32 → 64 → 96 → 160 → 320 kênh
- Conv cuối: 1×1 với 1280 kênh
- Global average pooling
- Lớp phân loại đầu ra (1×1 conv)

Từ bảng, ta thấy được MobileNetV2 bao gồm:

- Một lớp convolution đầu tiên với 32 filters.
- Sau đó là 19 block bottleneck residual, mỗi block gồm ba bước chính:
  - 1×1 conv + ReLU6: mở rộng số kênh từ k → tk (thường t = 6).
  - 3×3 depthwise conv + ReLU6: áp dụng lọc đặc trưng.
  - 1×1 conv (không có activation): nén số kênh xuống k'.
- Nếu đầu vào và đầu ra cùng kích thước, sẽ có kết nối tắt (residual)

1.2.2.5. Tham số thu gọn mô hình

Để MobileNet có thể trở nên gọn nhẹ hơn nữa, các tham số alpha (α) và rho (ρ) được đưa vào để làm giảm số kênh và độ phân giải của ảnh đầu vào [2].

Width Multiplier (α) làm mô hình gọn nhẹ hơn bằng cách giảm số kênh (channel) tại tất cả các lớp. Giá trị này thường nằm trong khoảng từ 0-1, điển hình là 1, 0.75, 0.5, 0.25. Chi phí khi áp dụng α là:

$$D_K . D_K . \alpha M . D_F . D_F + \alpha M . \alpha N . D_F . D_F$$

Resolution Multiplier (ρ) giảm độ phân giải của ảnh input, thường nằm trong khoảng 0-1, thường để giảm kích thước đầu vào xuống còn 224, 198, 160 và 128. Việc giảm độ phân giải có thể khiến mô hình bỏ sót các đặc điểm khuôn mặt. Chi phí khi áp dụng ρ là:

$$D_K . D_K . \alpha M . \rho D_F . \rho D_F + \alpha M . \alpha N . \rho D_F . \rho D_F$$

Hơn nữa, α và ρ có thể cùng được áp dụng để làm nhẹ mô hình hơn nữa:

$$D_K . D_K . M . \rho D_F . \rho D_F + M . N . \rho D_F . \rho D_F$$

Nhờ hai tham số này mà mô hình có thể được điều chỉnh sao cho phù hợp với tài nguyên của hệ thống, mà vẫn giữ được mức độ chính xác trong khoảng chấp nhận được.

Tuy nhiên, với α và ρ càng nhỏ thì độ chính xác của mô hình sẽ càng thấp.

Ngoài ra, nhóm tác giả còn thêm một tham số nữa là hệ số mở rộng (t) [8]. Hệ số này được sử dụng ở đầu vào, có nhiệm vụ làm tăng số kênh đầu vào lên t lần, mở rộng không gian, qua đó giúp mạng học được những đặc trưng phức tạp hơn. Trong MobileNetV2, hệ số t thường có giá trị 6.

1.2.3. Kiến trúc FaceNet sử dụng MobileNet

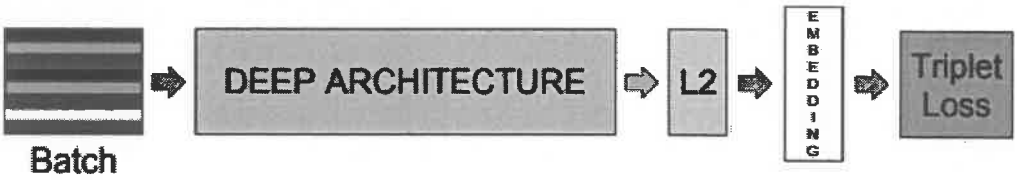
1.2.3.1. Tổng quan về FaceNet

FaceNet, được giới thiệu vào năm 2015, là một phương pháp đột phá trong nhận diện khuôn mặt, có ảnh hưởng sâu rộng đến cả nghiên cứu lẫn các ứng dụng thực tiễn trong công nghệ nhận diện khuôn mặt. FaceNet là mô hình học sâu có khả năng biểu diễn khuôn mặt bằng vector trong không gian Euclidean. Bằng cách sử

dụng MobileNet trong việc trích xuất được các đặc điểm khuôn mặt, FaceNet đạt được độ chính xác ấn tượng đồng thời vẫn duy trì hiệu quả tính toán. Hệ thống tối ưu hóa trực tiếp lên embedding thay vì các biểu diễn trung gian, tạo ra một framework thống nhất, trong đó các tác vụ như xác minh khuôn mặt (face verification), nhận dạng khuôn mặt (face recognition), và phân cụm (clustering) có thể được triển khai bằng cách sử dụng các embedding từ FaceNet như các vector đặc trưng (feature vectors).

Khi mới được công bố, FaceNet đã đạt độ chính xác kỷ lục là 99.63% trên bộ dữ liệu Labeled Faces in the Wild (LFW) và 95.12% trên YouTube Faces DB, giúp giảm tỷ lệ lỗi khoảng 30% so với các kết quả tốt nhất trước đó. Bước nhảy vọt về hiệu suất này cho thấy tính hiệu quả của việc tối ưu trực tiếp các embedding thay vì dựa vào các biểu diễn trung gian (intermediate representations).

1.2.3.2. Kiến trúc của FaceNet



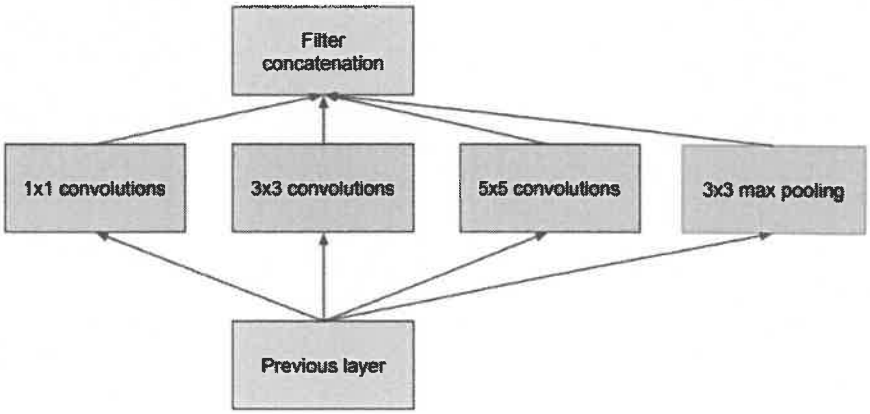
Hình 1.9. Kiến trúc của FaceNet [2]

Mạng FaceNet bao gồm một lớp đầu vào theo lô (batch) và một mạng tích chập sâu, sau đó là chuẩn hóa L2 để tạo các vector embedding và cuối cùng tính toán triplet loss để tạo khoảng cách giữa các đối tượng giống nhau càng nhỏ càng tốt, khoảng cách giữa các đối tượng khác nhau càng lớn càng tốt.

Nó sử dụng mạng nơ-ron tích chập sâu để học cách ánh xạ mỗi hình ảnh vào không gian embedding (Euclidean embedding space), và huấn luyện mạng sao cho khoảng cách L2 bình phương trong không gian này phản ánh trực tiếp độ tương tự khuôn mặt.

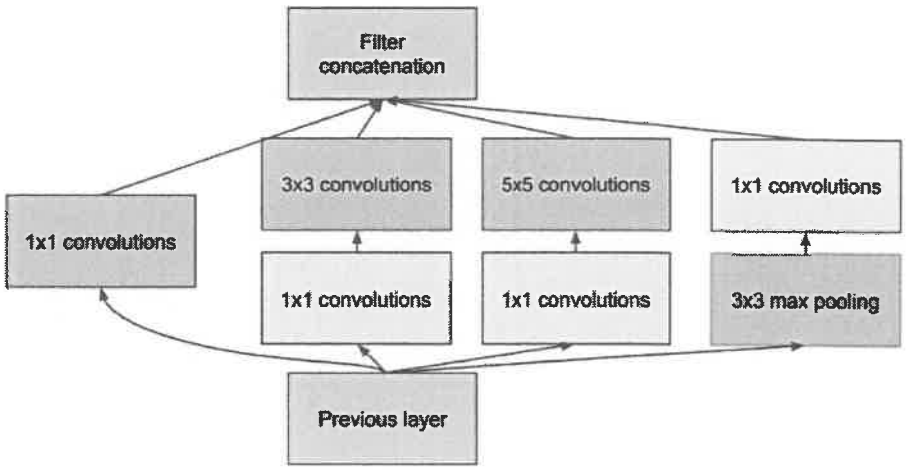
Mạng tích chập được sử dụng trong FaceNet là GoogleNet, sử dụng module Inception, vì vậy mạng này còn có tên gọi khác là mạng Inception.

Module Inception bao gồm các phép tích chập với các kích thước khác nhau, cụ thể là phép tích chập 1x1, 3x3 và 5x5 cùng với 1 lớp max pooling 3x3. Các đặc trưng thu được từ các lớp tích chập và lớp gộp này được tổng hợp lại với nhau thành đầu ra cuối cùng, cũng là đầu vào của mô-đun tiếp theo.



Hình 1.10. Inception module

Tuy nhiên, do số lượng lớn kernel được sử dụng, làm cho độ phức tạp tính toán tăng lên, dẫn đến số lượng đặc trưng bị hạn chế. GoogleNet đã sử dụng các phép tích chập 1x1 để tăng giảm các chiều của đặc trưng, giúp giảm chi phí tính toán.



Hình 1.11. Inception module giảm không gian

Toàn bộ mạng GoogleNet được tạo ra bằng cách xếp chồng các module Inception với nhau, tổng cộng là 22 lớp. Ở cuối mạng, một lớp average pooling được sử dụng thay thế lớp fully connected để tăng cường độ chính xác.

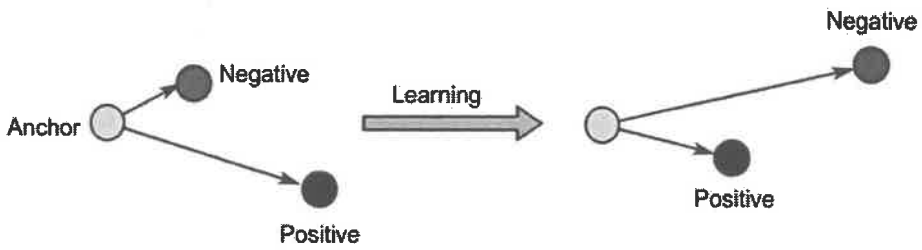
Để huấn luyện mô hình, FaceNet sử dụng hàm Triplet loss để học cách phân biệt rõ ràng các khuôn mặt với nhau [2]. Hàm Triplet loss hoạt động dựa trên bộ ba ảnh gồm:

- Anchor: ảnh gốc của một người
- Positive: ảnh khác cũng của người đó
- Negative: ảnh của người khác

. Hàm Triplet loss được biểu diễn như sau:

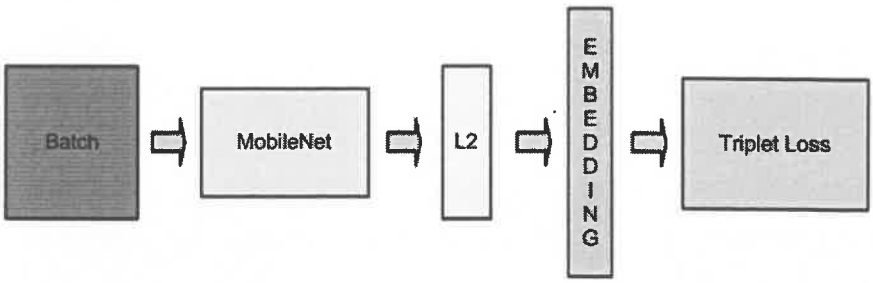
$$L = \sum_{i}^{N} \left[\left\| f(x_i^a) - f(x_i^p) \right\|_2^2 - \left\| f(x_i^a) - f(x_i^n) \right\|_2^2 + \alpha \right]_+$$

Hàm Triplet loss có nhiệm vụ rút ngắn khoảng cách giữa anchor và positive, và tăng khoảng cách cho negative [2]. Khoảng cách này càng ngắn thì chứng tỏ hai khuôn mặt càng giống nhau.



Hình 1.12. Minh họa hàm Triplet loss [2]

1.2.3.3. FaceNet dựa trên MobileNet



Hình 1.13. Kiến trúc FaceNet với backbone MobileNet [3]

Một điểm yếu của GoogleNet là kiến trúc của nó khá lớn, dẫn tới việc số lượng tính toán cần đến là rất nhiều, ảnh hưởng đến tốc độ xử lý của mô hình và không phù hợp với những thiết bị có phần cứng hạn chế. Với việc sử dụng MobileNet thay thế cho GoogleNet trong việc trích xuất đặc trưng khuôn mặt, số lượng tính toán được giảm đáng kể với việc ứng dụng depthwise separable convolution mà vẫn giữ được kết quả tốt. Với việc 95% thời gian tính toán dành cho các phép tích chập 1x1, mà các phép tính này chiếm tới 74% tổng số tham số của mô hình, tốc độ xử lý được giảm đi đáng kể do các phép này không cần phải tái sắp xếp bộ nhớ (memory rendering) và có thể thực hiện qua GEMM

CHƯƠNG 2: XÂY DỰNG HỆ THỐNG NHẬN DIỆN KHUÔN MẶT

2.1. Kiến trúc hệ thống

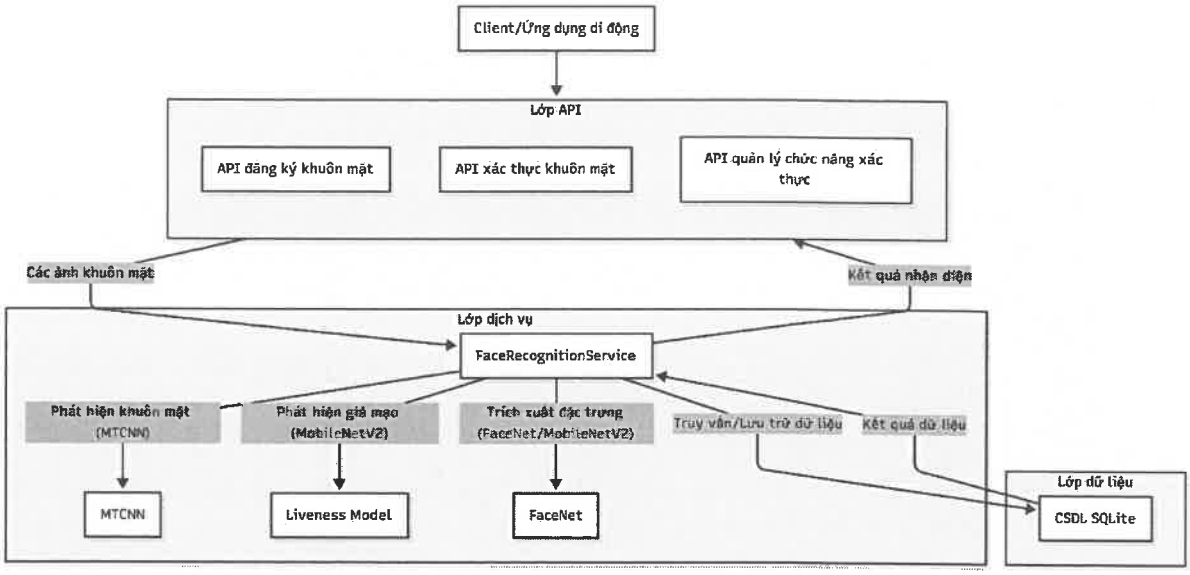
Hệ thống nhận diện khuôn mặt này là một giải pháp hữu hiệu được xây dựng bằng FastAPI và ứng dụng các công nghệ học sâu hiện đại. Chức năng chính của hệ thống là khả năng nhận diện khuôn mặt và nhận dạng khuôn mặt giả mạo chính xác thông qua RESTful API. Hệ thống kết hợp các giai đoạn phát hiện khuôn mặt, phát hiện khuôn mặt giả mạo và nhận dạng khuôn mặt để đảm bảo xác thực người dùng mạnh mẽ và an toàn.

Hệ thống được thiết kế theo kiến trúc phân tầng, giúp đảm bảo được tính mở rộng, dễ bảo trì và tối ưu hiệu năng. Các thành phần chính bao gồm:

- Lớp API (FastAPI): lớp này chịu trách nhiệm tiếp nhận yêu cầu từ người dùng, kiểm tra tính hợp lệ của đầu vào, sau đó khi hệ thống xử lý xong, trả lại kết quả cho người dùng.
- Lớp dịch vụ (FaceRecognitionService): lớp này chứa các logic xử lý khuôn mặt, bao gồm xử lý ảnh, phát hiện khuôn mặt, nhận diện khuôn mặt giả mạo và xác thực khuôn mặt.
- Lớp dữ liệu (SQLAlchemy): quản lý thông tin của người dùng và các vector embedding khuôn mặt trong cơ sở dữ liệu SQLite.

Mô hình này giúp phân biệt rõ ràng giữa các vai trò, tăng khả năng kiểm soát và mở rộng sau này.

Dưới đây là sơ đồ kiến trúc hệ thống:



Hình 2.1. Sơ đồ khối kiến trúc hệ thống

2.2. Quy trình xử lý

Quy trình xử lý nhận diện khuôn mặt trong hệ thống được chia thành ba giai đoạn chính:

- Phát hiện khuôn mặt (Face Detection)

Hệ thống sử dụng MTCNN (Multi-task Cascaded Convolutional Neural Network) để phát hiện vị trí khuôn mặt trong ảnh. MTCNN là một mô hình mạng tích chập được thiết kế đặc biệt cho bài toán phát hiện khuôn mặt đa giai đoạn, kết hợp giữa định vị và hiệu chỉnh các điểm đặc trưng trên khuôn mặt.

- Kiểm tra giả mạo khuôn mặt (Liveness Detection)

Để ngăn chặn các hành vi tấn công giả mạo bằng sử dụng ảnh của người dùng, hệ thống tích hợp một mô hình học sâu được huấn luyện riêng cho tác vụ phân biệt khuôn mặt thật và giả mạo. Mô hình này học được các đặc trưng để đưa ra quyết định chính xác.

- Nhận diện khuôn mặt (Face Recognition)

Sau khi xác thực khuôn mặt là "sống", hệ thống sử dụng mô hình FaceNet với backbone là MobileNetV2, được chuyển đổi sang định dạng TensorFlow Lite để tăng hiệu năng, nhằm trích xuất vector đặc trưng (embedding) cho khuôn mặt. Sau đó,

vector này được so sánh với cơ sở dữ liệu bằng cách tính khoảng cách cosine để xác định danh tính người dùng.

2.3. Cơ sở dữ liệu của hệ thống

Cơ sở dữ liệu của hệ thống gồm một bảng duy nhất là "users", với nhiệm vụ lưu lại các thông tin cần thiết cho chức năng nhận diện khuôn mặt.

Bảng gồm có trường id là khóa chính, trường tên, ba trường để lưu trữ vector embedding dạng nhị phân theo các hướng nhìn thẳng, quay trái, quay phải. Ngoài ra, còn một trường dạng boolean, được sử dụng để kiểm soát việc bật/tắt chức năng nhận diện khuôn mặt cho ứng dụng.

Kết nối cơ sở dữ liệu được quản lý thông qua phương pháp dựa trên phiên, với việc xử lý kết nối phù hợp và các cân nhắc về an toàn luồng. Thiết lập cơ sở dữ liệu tuân theo các biện pháp thực hành tốt nhất của SQLAlchemy với các mô hình cơ sở khai báo và hệ thống quản lý phiên đảm bảo dọn dẹp tài nguyên phù hợp.

Bảng 2.1. Bảng users của cơ sở dữ liệu

users	
id	PK, String
name	String
face_front_embedding	LargeBinary
face_left_embedding	LargeBinary
face_right_embedding	LargeBinary
isEnableFaceId	Boolean

2.4. Các chức năng chính của hệ thống

2.4.1. Đăng ký khuôn mặt

Để có thể nhận diện được khuôn mặt thì bước đầu tiên cần phải có dữ liệu khuôn mặt của người dùng để tiến hành so sánh với đầu vào gồm ba ảnh chứa khuôn

mặt của người dùng theo các hướng: nhìn thẳng, quay trái, quay phải và id của chúng. Để có dữ liệu khuôn mặt, hệ thống sẽ trích xuất các vector embedding khuôn mặt từ những ảnh này. Hệ thống sau đó sẽ kiểm tra xem dữ liệu khuôn mặt có tồn tại trong cơ sở dữ liệu hay không, nếu có thì thay thế chúng với những dữ liệu mới thu được, còn nếu không thì sẽ tạo mới.

2.4.2. Nhận diện khuôn mặt

Hệ thống được tối ưu để xử lý ảnh hoặc video đầu vào gần như ngay lập tức, đáp ứng yêu cầu xác thực trong các ứng dụng thực tế, ví dụ như điểm danh, mở khóa thiết bị, hoặc kiểm soát truy cập.

Từ đầu vào là ảnh, hệ thống phát hiện các khuôn mặt có trong ảnh, sau đó trích xuất các vector embedding với từng khuôn mặt. Sau đó, hệ thống so sánh với khuôn mặt tương ứng với người dùng trong cơ sở dữ liệu bằng hàm tương đồng côsin, và trả về kết quả cho người dùng.

2.4.3. Nhận diện khuôn mặt giả mạo

Phát hiện giả mạo khuôn mặt (Liveness detection) là công nghệ quan trọng trong xác thực sinh trắc học, nhằm phân biệt dữ liệu thật với các hình thức tấn công giả mạo như in ảnh, video replay, hoặc kính áp tròng có họa tiết. Trong bối cảnh các hệ thống nhận dạng khuôn mặt, vân tay, giọng nói ngày càng phổ biến, liveness detection đóng vai trò then chốt để ngăn chặn xâm nhập trái phép, đảm bảo độ tin cậy và bảo mật.

Hệ thống của tôi sử dụng mô hình MobileNetV2 để xử lý phát hiện khuôn mặt giả mạo. Nó được chọn vì những ưu điểm sau:

- Với kiến trúc gồm các tích chập phân tách theo chiều sâu (depthwise separable convolution) giúp làm giảm số lượng tham số và phép tính, việc sử dụng kiến trúc inverted residual với linear bottleneck giúp cải thiện hiệu năng và khả năng truyền thông tin mà vẫn giữ được độ chính xác. Việc giảm số lượng tham số và phép

tính giúp mô hình trở nên nhẹ hơn, giúp nó có thể được triển khai trên các thiết bị có tài nguyên hạn chế

- Mặc dù MobileNetV2 sử dụng ít tham số và phép tính hơn, nhưng độ chính xác của mô hình vẫn được đảm bảo. Hiệu suất nhận diện hình ảnh tốt, đã được kiểm chứng trên nhiều tập dữ liệu lớn, đảm bảo độ chính xác cao cho bài toán nhận diện khuôn mặt giả mạo.

- Dễ dàng tích hợp và mở rộng nhờ thiết kế module hóa, thuận tiện cho việc áp dụng transfer learning để thích nghi với các bài toán sinh trắc học khác nhau. Hệ thống sử dụng mô hình MobileNetV2 đã qua huấn luyện trên tập ImageNet, sau đó thực hiện đóng băng (frozen) giúp tránh việc làm hỏng các trọng số đã học, kế thừa được khả năng trích xuất các đặc trưng hình ảnh cấp thấp và trung bình đã được học từ hàng triệu ảnh. Để giúp mô hình có thể phân biệt được ảnh thật hay giả, lớp phân loại cuối cùng của MobileNetV2 được thay thế bằng bộ phân loại tùy chỉnh.

Nhờ các đặc điểm này, MobileNetV2 giúp cân bằng giữa tốc độ, độ chính xác và khả năng triển khai thực tế trên nhiều nền tảng, là lựa chọn tối ưu cho liveness detection trên thiết bị giới hạn tài nguyên.

2.4.3.1. So sánh với các mô hình khác

Ngoài MobileNetV2, chúng ta có thể lựa chọn các mô hình khác như ResNet, EfficientNet. Các mô hình này đều là những mô hình đã chứng minh được sự hiệu quả trong việc xử lý hình ảnh.

Mô hình ResNet

ResNet là Mô hình kiến trúc mạng nơ-ron được phát triển bởi các nhà nghiên cứu của Microsoft Research năm 2015. ResNet đã giải quyết được vấn đề "thoái hóa" (degradation) - khi độ sâu mạng tăng, độ chính xác bão hòa rồi giảm nhanh chóng, không phải do hiện tượng overfitting mà do khó khăn trong tối ưu hóa. Từ đó, ResNet có thể có tối đa 152 lớp [10].

ResNet đưa ra mô hình học Residual Learning giúp việc tối ưu hóa các mạng rất sâu trở nên dễ dàng hơn, đồng thời giúp tăng độ chính xác khi tăng số tầng mạng mà không gặp phải vấn đề suy giảm (degradation):

- Thay vì học trực tiếp hàm mục tiêu, residual learning học hàm phần dư $F_x = H_x - x$, từ đó dễ dàng hơn để mạng tối ưu hóa. Đầu ra sẽ trở thành $F_x + x$
- Khắc phục hiện tượng biến mất gradient (vanishing gradient) nhờ các đường kết nối tắt (shortcut connections), giúp gradient truyền tải hiệu quả qua nhiều tầng sâu [10].

Kiến trúc cơ bản: ResNet được xây dựng từ các khối residual block

- Residual Block: Khối xây dựng cơ bản có công thức y = F(x, {Wi}) + x, trong đó x là đầu vào, y là đầu ra, và F(x, {Wi}) là ánh xạ tàn dư (residual mapping) cần học.
- Skip Connections: Kết nối tắt (shortcut connections) thực hiện ánh xạ đồng nhất, cộng đầu ra với đầu vào qua phép cộng theo từng phần tử.
- Bottleneck Architecture: Cho các mạng sâu hơn (50, 101, 152 lớp), ResNet sử dụng thiết kế thắt cổ chai với 3 lớp: 1×1, 3×3, và 1×1 convolution.
- Khi số chiều đầu vào và đầu ra của block không bằng nhau, ResNet dùng phép chiếu tuyến tính 1×1 convolution để điều chỉnh chiều nhằm thực hiện phép cộng [10].

Kết quả trên ImageNet của ResNet: [12]

- ResNet-34: Top-1 error 25.03%, Top-5 error 7.76%
- ResNet-50: Top-1 error 22.85%, Top-5 error 6.71%
- ResNet-152: Top-1 error 21.43%, Top-5 error 5.71%

Ưu điểm của ResNet:

- Giải quyết được vấn đề thoái hóa, cho phép huấn luyện mạng rất sâu (lên đến 152 lớp)
- Cải thiện độ chính xác đáng kể khi tăng độ sâu

Tuy nhiên, khi so sánh với MobileNetV2, ResNet bộc lộ nhiều khuyết điểm:

- Hiệu quả tính toán: MobileNetV2 vượt trội với chỉ 300M FLOPs so với 4.1B FLOPs của ResNet-50, giảm hơn 13 lần chi phí tính toán [12].
- Kích thước mô hình: MobileNetV2 nhỏ hơn đáng kể với 3.4M parameters so với 25.6M của ResNet-50 [12]
- Độ chính xác: ResNet-50 đạt độ chính xác cao hơn (76.0% vs 72.0% top-1 accuracy), nhưng chênh lệch chỉ 4%, có thể chấp nhận được cho nhiều ứng dụng thực tế [12].

Từ đó, ta thấy được khi cần triển khai trên thiết bị di động, edge devices, hoặc các môi trường hạn chế tài nguyên thì MobileNetV2 cung cấp sự cân bằng tốt hơn giữa độ chính xác và hiệu quả, với tốc độ nhanh hơn 2-4 lần và sử dụng tài nguyên ít hơn đáng kể

Mô hình EfficientNet

EfficientNet là một họ mạng nơ-ron tích chập (CNN) được giới thiệu bởi nhóm nghiên cứu Google vào năm 2019, đặc biệt nổi bật với phương pháp compound scaling (mở rộng kết hợp) cách mạng. Mô hình này đạt được hiệu suất vượt trội bằng cách cân bằng đồng thời ba chiều quan trọng của mạng nơ-ron: độ sâu (depth), độ rộng (width) và độ phân giải (resolution) [11].

Kiến trúc mô hình bao gồm: [11]

- 9 giai đoạn chính với các khối building block khác nhau
- Khối MBConv (Mobile Inverted Bottleneck Convolution) làm thành phần cốt lõi

- Squeeze-and-Excitation optimization được tích hợp để cải thiện hiệu suất
- Tổng cộng 53 lớp với kiến trúc tối ưu cho thiết bị di động

MBConv là xương sống của EfficientNet, lấy cảm hứng và kế thừa từ MobileNetV2. Đặc trưng của MBConv là sử dụng cấu trúc inverted bottleneck và depthwise separable convolution:

- Inverted bottleneck:

Khối này mở rộng số kênh từ đầu vào hẹp ra rộng bằng một convolution 1×1 (expansion), sau đó áp dụng depthwise convolution, cuối cùng thu hẹp trở lại bằng một convolution 1×1 (projection). Khác với residual bottleneck truyền thống (rộng → hẹp → rộng), MBConv đi theo hướng ngược lại: hẹp → rộng → hẹp.

- Depthwise separable convolution:

Chia hoạt động convolution thành hai bước:

  - Depthwise convolution áp dụng một bộ lọc cho mỗi kênh đầu vào riêng rẽ (giảm rất nhiều số lượng tham số và tính toán).
  - Pointwise convolution (1×1 convolution) tổng hợp các kênh lại để tạo đặc trưng mới.

- Squeeze-and-Excitation (SE) optimization:

MBConv trong EfficientNet được tăng cường với khối SE để học trọng số quan trọng ở từng kênh, nâng cao hiệu suất học đặc trưng.

Phương pháp Compound Scaling là đóng góp quan trọng nhất của EfficientNet. Thay vì mở rộng một chiều đơn lẻ, phương pháp này sử dụng công thức:

- Depth: $d = \alpha^{\varphi}$
- Width: $w = \beta^{\varphi}$
- Resolution: $r = \gamma^{\varphi}$ [11]

Với ràng buộc: $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$, đảm bảo FLOPS tăng khoảng 2^{φ} lần. [11]

Các hệ số tối ưu cho EfficientNet-B0 là: $\alpha = 1.2$, $\beta = 1.1$, $\gamma = 1.15$. [11]

EfficientNet đạt được các kết quả khá tốt: 77.1% trên tập dữ liệu ImageNet. Ngoài ra, EfficientNet cũng thể hiện khả năng transfer learning xuất sắc, đạt state-of-the-art trên 5/8 dataset phổ biến với trung bình 9.6x ít tham số hơn các mô hình khác [11].

Mặc dù EfficientNet có hiệu suất tổng thể cao hơn, MobileNetV2 vẫn cho thấy được hiệu quả tính toán cao hơn MobileNetV2 chỉ có 3.4 triệu tham số so với 5.3 triệu của EfficientNet-B0, phù hợp với các thiết bị có tài nguyên hạn chế.

2.4.4. Quản lý người dùng và bật/tắt nhận diện khuôn mặt

Hệ thống cho phép tạo, cập nhật, xóa người dùng và có thể bật hoặc tắt tính năng nhận diện khuôn mặt cho từng cá nhân, hỗ trợ nhiều mức độ bảo mật linh hoạt.

2.5. Hiệu suất và khả năng mở rộng

Đánh giá về hiệu suất và khả năng mở rộng của hệ thống:

- Hệ thống sử dụng FastAPI – một framework nhẹ và hiệu suất cao – giúp xử lý hàng trăm yêu cầu mỗi giây.
- Việc sử dụng TensorFlow Lite và SQLite giúp hệ thống có thể triển khai trên các thiết bị edge hoặc hệ thống nhúng.
- Kiến trúc phân tầng giúp dễ dàng mở rộng với các tính năng mới như nhận diện biểu cảm, phân tích giới tính, độ tuổi,...

CHƯƠNG 3: TRIỂN KHAI VÀ THỬ NGHIỆM

3.1. Huấn luyện mô hình

3.1.1. Nhận diện khuôn mặt

Hệ thống nhận diện khuôn mặt hiện đại đòi hỏi sự kết hợp giữa độ chính xác cao và khả năng hoạt động thời gian thực trên phần cứng hạn chế. Nghiên cứu này đề xuất một kiến trúc lai sử dụng MTCNN (Multi-Task Cascaded Convolutional Networks) cho giai đoạn phát hiện khuôn mặt và Fast FaceNet dựa trên MobileNetV2 cho trích xuất đặc trưng, tạo nên một pipeline end-to-end tối ưu cho thiết bị biên.

3.1.1.1. Giai đoạn phát hiện khuôn mặt với MTCNN tối ưu

MTCNN hoạt động thông qua ba mạng con xếp tầng (P-Net, R-Net, O-Net), mỗi mạng giải quyết các nhiệm vụ cụ thể với độ phức tạp tăng dần. P-Net (Proposal Network) xử lý ảnh đầu vào 12×12 pixel để đề xuất các vùng khuôn mặt tiềm năng thông qua một mạng CNN nhẹ sử dụng depthwise separable convolution. Công thức tính confidence score tại P-Net được biểu diễn:

$$\mathrm{sp} = \sigma(\mathrm{Wp} * \mathrm{X} + \mathrm{bp})$$

trong đó Wp là kernel tích chập 3×3 với stride 2, giảm 75% tham số so với kiến trúc gốc. R-Net (Refine Network) tiếp nhận các proposal từ P-Net, thực hiện regression để tinh chỉnh bounding box:

$$\Delta\mathrm{x} = \mathrm{Wr} * \mathrm{X} + \mathrm{br}$$

với $\mathrm{Wr} \in \mathrm{R}^{3\times3\times32\times64}$. Giai đoạn cuối cùng, O-Net (Output Network), tích hợp cơ chế spatial attention tập trung vào các vùng quan trọng như mắt và mũi:

$$\mathrm{Aattention} = \mathrm{softmax}(\mathrm{Conv1} \times 1(\mathrm{Fmid}))$$

$$\mathrm{Fout} = \mathrm{Aattention} \odot \mathrm{Fmid}$$

Để tối ưu hóa cho thiết bị IoT, phiên bản MTCNN cải tiến sử dụng 8-bit quantization và tiling strategy, chia ảnh lớn thành các mảnh 640×480 pixel xử lý song song. Kết quả thử nghiệm trên tập dữ liệu WIDER Face cho thấy độ chính xác 93.8%

với thời gian xử lý chỉ 35ms/ảnh trên Raspberry Pi 4, tiêu thụ năng lượng giảm 67% so với bản gốc.

3.1.1.2. Trích xuất đặc trưng với Fast FaceNet

Thay thế Inception-ResNet bằng MobileNetV2 trong FaceNet mang lại những cải tiến đáng kể về hiệu suất. Kiến trúc MobileNetV2 gồm 15 inverted residual blocks với hệ số mở rộng t = 6 trong đó mỗi block áp dụng công thức:

$$\text{Fout} = \text{Proj}(\text{ReLU6}(\text{DWConv3} \times 3(\text{Exp}(\text{X})))) + \text{Xskip}$$

với $\text{Exp}(\text{X}) = \text{Conv1} \times 1(\text{X})$ mở rộng số kênh lên 6 lần. Lớp projection cuối cùng sử dụng linear activation để tránh mất thông tin trong không gian chiều thấp. Quá trình huấn luyện kết hợp triplet loss.

3.1.1.3. Tích hợp hệ thống end-to-end

Luồng xử lý hoàn chỉnh bao gồm:

- Tiền xử lý ảnh: Chuẩn hóa histogram và cân bằng ánh sáng sử dụng CLAHE.
- Phát hiện đa phương thức: Chạy MTCNN trên ảnh RGB.
- Trích xuất đặc trưng: Fast FaceNet xử lý hình ảnh để lấy được vector embedding khuôn mặt.
- Matching thông minh: sử dụng cosine similarity.

3.1.2. Phát hiện khuôn mặt giả mạo

Trong đề án này, tôi phát triển một mô hình học sâu nhằm phát hiện ảnh khuôn mặt thật/giả, một thành phần quan trọng trong hệ thống xác thực sinh trắc học. Mô hình được xây dựng dựa trên phương pháp học chuyển tiếp (transfer learning) với kiến trúc MobileNetV2, được huấn luyện và đánh giá trên một tập dữ liệu quy mô lớn. Nội dung sẽ bao gồm mô tả bài toán, phân tích tập dữ liệu, trình bày kiến trúc mô hình, chiến lược huấn luyện và cuối cùng là các kết quả thực nghiệm đạt được.

3.1.2.1. Mô tả bài toán

Bài toán phát hiện giả mạo khuôn mặt được định nghĩa là một bài toán phân loại nhị phân. Cho một ảnh đầu vào I, mô hình cần phải dự đoán nhãn y của ảnh đó, trong đó:

- y = 1 nếu ảnh I chứa khuôn mặt của người thật (live).
- y = 0 nếu ảnh I là một phiên bản giả mạo (spoof).

Mục tiêu là xây dựng một bộ phân loại f_I sao cho đầu ra của nó có thể được sử dụng để xác định nhãn y với độ chính xác cao nhất có thể.

3.1.2.2. Mô tả tập dữ liệu

Để huấn luyện và đánh giá mô hình, luận án sử dụng tập dữ liệu CelebA-Spoof, một bộ dữ liệu công khai và quy mô lớn được thiết kế chuyên biệt cho bài toán này.

- Quy mô: Tập dữ liệu bao gồm 100.000 hình ảnh được gán nhãn, đảm bảo đủ lớn để huấn luyện các mô hình học sâu phức tạp và tránh hiện tượng học vẹt (overfitting).
- Phân bố lớp: Dữ liệu được chia thành hai lớp: "live" và "spoof". Dựa trên phân tích trong quá trình xử lý, tập dữ liệu có sự mất cân bằng nhẹ giữa các lớp. Cụ thể, lớp "spoof" (nhãn 0) có số lượng mẫu ít hơn lớp "live" (nhãn 1). Điều này được thể hiện qua trọng số lớp tính toán được trong quá trình huấn luyện: trọng số của lớp 0 là khoảng 1.5 trong khi của lớp 1 là 0.75, cho thấy mô hình cần chú trọng hơn vào việc học các mẫu thuộc lớp thiểu số.
- Đặc điểm ảnh: Các ảnh trong tập dữ liệu có độ phân giải và điều kiện ánh sáng đa dạng, mô phỏng các tình huống sử dụng trong thực tế. Các tấn công giả mạo bao gồm nhiều loại hình tấn công trình diễn phổ biến, giúp mô hình có khả năng tổng quát hóa tốt hơn.
- Phân chia dữ liệu: Tập dữ liệu được phân chia thành hai tập con: tập huấn luyện (training set) và tập kiểm định (validation set) theo tỷ lệ 80/20. Quá trình chia được thực hiện theo phương pháp phân chia có phân tầng (stratified splitting) dựa trên nhãn của dữ liệu. Điều này đảm bảo rằng tỷ lệ mẫu giữa hai lớp "live" và "spoof" trong tập huấn luyện và tập kiểm định là tương đương nhau. Cụ thể, tập huấn luyện bao gồm 80.000 ảnh và tập kiểm định có 20.000 ảnh.

3.1.2.3. Tiền xử lý và tăng cường dữ liệu

Quá trình tiền xử lý và tăng cường dữ liệu là một bước quan trọng, ảnh hưởng trực tiếp đến hiệu năng của mô hình. Trong dự án này, một quy trình xử lý dữ liệu hiệu quả đã được xây dựng bằng thư viện TensorFlow:

- Chuẩn hóa kích thước và giá trị pixel: Tất cả các ảnh đầu vào, dù có kích thước ban đầu khác nhau, đều được đồng nhất về kích thước 224x224 pixels. Đây là kích thước đầu vào tiêu chuẩn của kiến trúc MobileNetV2. Giá trị của mỗi pixel sau đó được chuẩn hóa về khoảng [0, 1] bằng cách chia cho 255.
- Tăng cường dữ liệu (Data Augmentation): Để tăng tính đa dạng của dữ liệu huấn luyện và giúp mô hình có khả năng tổng quát hóa tốt hơn, các phép biến đổi ngẫu nhiên được áp dụng lên ảnh trong quá trình huấn luyện. Các kỹ thuật được sử dụng bao gồm:
    - Lật ngang ngẫu nhiên (Random Horizontal Flip): Mô phỏng góc nhìn khác nhau.
    - Xoay ngẫu nhiên (Random Rotation): Thay đổi góc nghiêng của khuôn mặt (giới hạn trong khoảng 0.1 radian).
    - Phóng to/thu nhỏ ngẫu nhiên (Random Zoom): Thay đổi khoảng cách từ camera đến khuôn mặt.
    - Thay đổi độ sáng ngẫu nhiên (Random Brightness): Tăng khả năng chống chịu với các điều kiện ánh sáng khác nhau.
    - Thay đổi độ tương phản ngẫu nhiên (Random Contrast): Giúp mô hình tập trung vào các đặc trưng kết cấu bền vững hơn.
- Xây dựng đường ống dữ liệu (Data Pipeline): Toàn bộ quá trình được đóng gói bằng tf.data.Dataset, một API hiệu năng cao của TensorFlow. Đường ống này cho phép đọc, tiền xử lý, tăng cường, và tạo các lô (batch) dữ liệu một cách song song và hiệu quả. Kích thước mỗi lô được thiết lập là 128. Kỹ thuật prefetch cũng được sử dụng để tải trước dữ liệu cho bước huấn luyện tiếp theo trong khi GPU đang xử lý lô hiện tại, giúp tối ưu hóa thời gian huấn luyện.

3.1.2.4. Kiến trúc mô hình

Mô hình được xây dựng dựa trên phương pháp học chuyển tiếp, tận dụng kiến thức đã được học từ một mô hình lớn trên một tập dữ liệu khổng lồ.

- Mô hình nền (Base Model): Kiến trúc MobileNetV2 được chọn làm mô hình nền. Đây là một kiến trúc CNN hiện đại, được tối ưu hóa cho hiệu suất cao trên các thiết bị có tài nguyên tính toán hạn chế như điện thoại di động. Các trọng số của mô hình này đã được huấn luyện trước trên tập dữ liệu ImageNet. Bằng cách sử dụng mô hình nền này, chúng ta kế thừa được khả năng trích xuất các đặc trưng hình ảnh cấp thấp và trung bình (như cạnh, góc, kết cấu) đã được học từ hàng triệu ảnh. Lớp phân loại cuối cùng (top layer) của MobileNetV2 được loại bỏ để thay thế bằng một bộ phân loại tùy chỉnh.
- Bộ phân loại tùy chỉnh (Custom Classifier Head): Các đặc trưng được trích xuất từ mô hình nền được đưa qua một chuỗi các lớp mới được thêm vào:
    - GlobalAveragePooling2D: Lớp này tính toán giá trị trung bình của từng bản đồ đặc trưng, giúp giảm đáng kể số lượng tham số và chống học vẹt.
    - Dense(1024, activation = 'relu'): Một lớp kết nối đầy đủ với 1024 nơ-ron và hàm kích hoạt ReLU để học các tổ hợp đặc trưng cấp cao.
    - Dropout(0.5): Lớp dropout với tỷ lệ 50% được thêm vào để chính quy hóa (regularization), ngẫu nhiên vô hiệu hóa một nửa số nơ-ron trong quá trình huấn luyện để ngăn chặn sự phụ thuộc quá mức vào một số đặc trưng nhất định.
    - Dense(1, activation = 'linear'): Lớp đầu ra cuối cùng với một nơ-ron duy nhất và hàm kích hoạt tuyến tính. Việc sử dụng hàm linear thay vì sigmoid là một lựa chọn kỹ thuật để kết hợp với hàm mất mát BinaryCrossentropy(from_logits = True), giúp tăng tính ổn định số học trong quá trình huấn luyện.

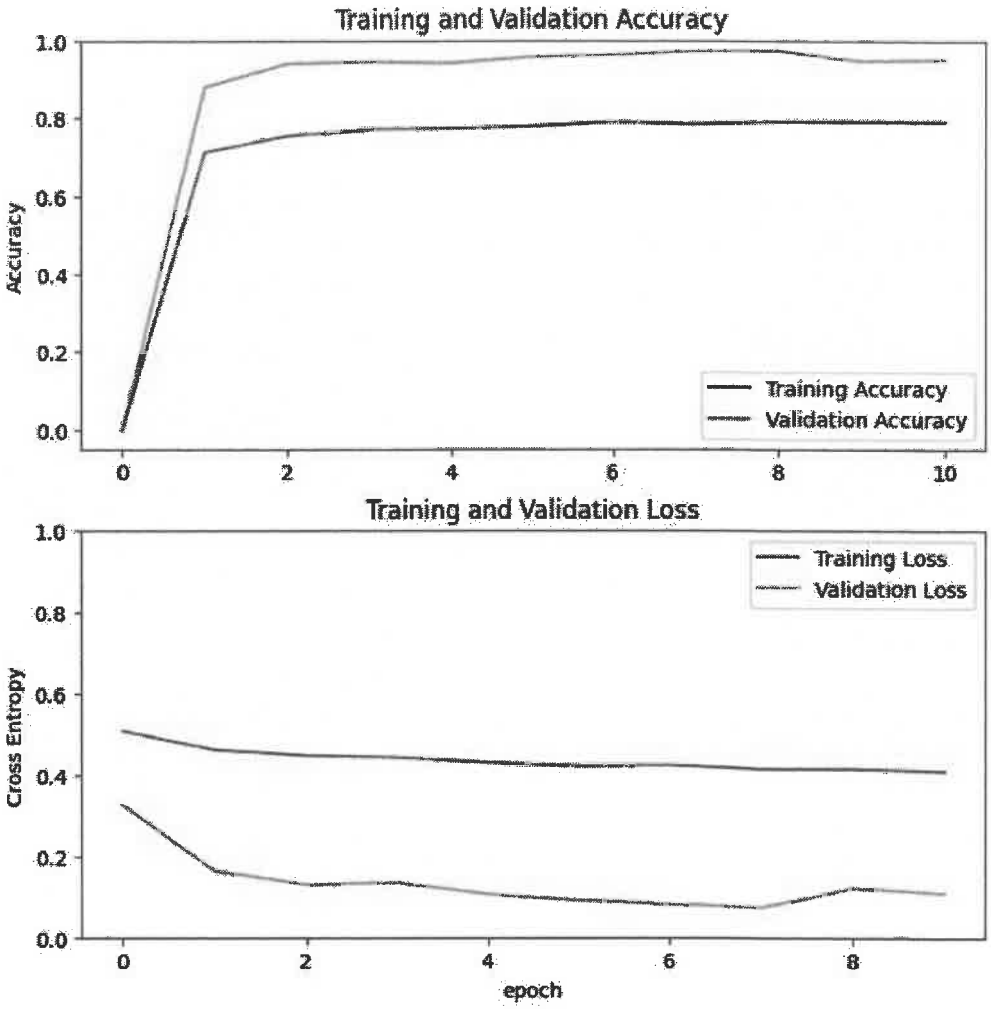
Tổng số tham số của mô hình hoàn chỉnh là khoảng 3.57 triệu, trong đó phần lớn đến từ các lớp tích chập của MobileNetV2 và lớp Dense tùy chỉnh.

3.1.2.5. Chiến lược huấn luyện và tinh chỉnh

- Tinh chỉnh (Fine-tuning): Để thích ứng mô hình với bài toán cụ thể, chiến lược tinh chỉnh được áp dụng. Thay vì huấn luyện lại toàn bộ mạng, 100 lớp đầu tiên của MobileNetV2 được "đóng băng" (frozen), tức là các trọng số của chúng không được cập nhật. Các lớp còn lại và bộ phân loại tùy chỉnh được "mở băng" để huấn luyện. Chiến lược này giúp giữ lại các đặc trưng chung đã học được (ở các lớp dưới) và chỉ điều chỉnh các đặc trưng chuyên biệt hơn (ở các lớp trên) cho phù hợp với dữ liệu giả mạo khuôn mặt.
- Hàm mất mát và Trình tối ưu hóa:
    - Hàm mất mát (Loss Function): mô hình sử dụng hàm mất mát BinaryCrossentropy(from_logits=True), phù hợp với bài toán phân loại nhị phân và kiến trúc đầu ra của mô hình.
    - Trình tối ưu hóa (Optimizer): AdamW (Adam with Weight Decay) được chọn với tốc độ học (learning rate) khá nhỏ là 10−4. AdamW là một biến thể cải tiến của Adam, thường mang lại kết quả hội tụ tốt hơn. Tốc độ học thấp là cần thiết trong quá trình tinh chỉnh để tránh phá vỡ các trọng số đã được huấn luyện trước.
- Xử lý mất cân bằng dữ liệu: Để giải quyết vấn đề mất cân bằng nhẹ của tập dữ liệu, trọng số lớp (class weights) đã được tính toán và áp dụng trong quá trình huấn luyện. Phương pháp này "phạt" nặng hơn những lỗi dự đoán sai trên lớp thiểu số (lớp "spoof"), buộc mô hình phải chú ý học các đặc trưng của lớp này hơn.
- Quá trình huấn luyện: Mô hình được huấn luyện trong 10 chu kỳ (epochs) với kích thước lô là 128. Độ chính xác (accuracy) được sử dụng làm thước đo đánh giá hiệu năng trên cả tập huấn luyện và tập kiểm định sau mỗi kỷ nguyên.
- Lớp Dense output với 1 đơn vị (phân loại nhị phân)

3.1.2.6. Đánh giá và phân tích

Quá trình huấn luyện mô hình đã được theo dõi chặt chẽ thông qua các giá trị mất mát và độ chính xác trên tập huấn luyện và tập kiểm định.



Hình 3.1. Biểu đồ độ chính xác và mất mát trong quá trình huấn luyện

Độ chính xác (Accuracy): Như thể hiện trên Hình 3.1 (phía trên), độ chính xác trên tập huấn luyện (Training Accuracy) tăng đều và ổn định qua các kỷ nguyên, bắt đầu từ khoảng 68% và đạt gần 80% ở cuối quá trình. Quan trọng hơn, độ chính xác trên tập kiểm định (Validation Accuracy) tăng rất nhanh trong các kỷ nguyên đầu, từ 87.8% ở kỷ nguyên 1 lên đến đỉnh điểm là 97.29% ở kỷ nguyên thứ 8. Mặc dù có sự sụt giảm nhẹ ở hai kỷ nguyên cuối, kết quả này cho thấy mô hình có khả năng phân biệt rất tốt giữa khuôn mặt thật và giả mạo.

Hàm mất mát (Loss): Biểu đồ mất mát (Hình 3.1, phía dưới) cho thấy giá trị mất mát trên cả hai tập đều giảm nhanh. Giá trị mất mát trên tập kiểm định (Validation

Loss) giảm xuống mức rất thấp (khoảng 0.073 ở chu kỳ 8), tương ứng với thời điểm độ chính xác đạt cao nhất.

Phân tích kết quả:

- Mô hình hội tụ nhanh và đạt được hiệu năng cao, chứng tỏ sự hiệu quả của phương pháp học chuyển tiếp và kiến trúc MobileNetV2.
- Khoảng cách giữa đường cong huấn luyện và đường cong kiểm định không quá lớn, cho thấy các kỹ thuật chính quy hóa như Dropout và tăng cường dữ liệu đã hoạt động hiệu quả trong việc ngăn chặn học vẹt.
- Độ chính xác trên 97% trên tập kiểm định là một kết quả rất khả quan, cho thấy mô hình có tiềm năng ứng dụng cao trong các hệ thống xác thực sinh trắc học thực tế.

3.2. Triển khai hệ thống nhận diện

Để cho phép truy cập service từ bên ngoài mạng cục bộ, chúng tôi sử dụng ngrok để tạo một tunnel an toàn. Ngrok cung cấp một static domain miễn phí, cho phép người dùng truy cập API từ bất kỳ đâu mà không cần cấu hình phức tạp về mạng.

Quy trình triển khai với ngrok:

- Tạo tài khoản và cài đặt ngrok: Đăng ký tài khoản ngrok và cài đặt công cụ dòng lệnh.
- Xác thực ngrok: Liên kết ngrok với tài khoản.
- Tạo static domain: Trong bảng điều khiển ngrok, tạo một static domain.
- Khởi động tunnel: Sử dụng static domain để tạo tunnel đến service FastAPI cục bộ.
- Truy cập API: Sử dụng URL https://your-static-domain.ngrok-free.app để truy cập API từ xa.

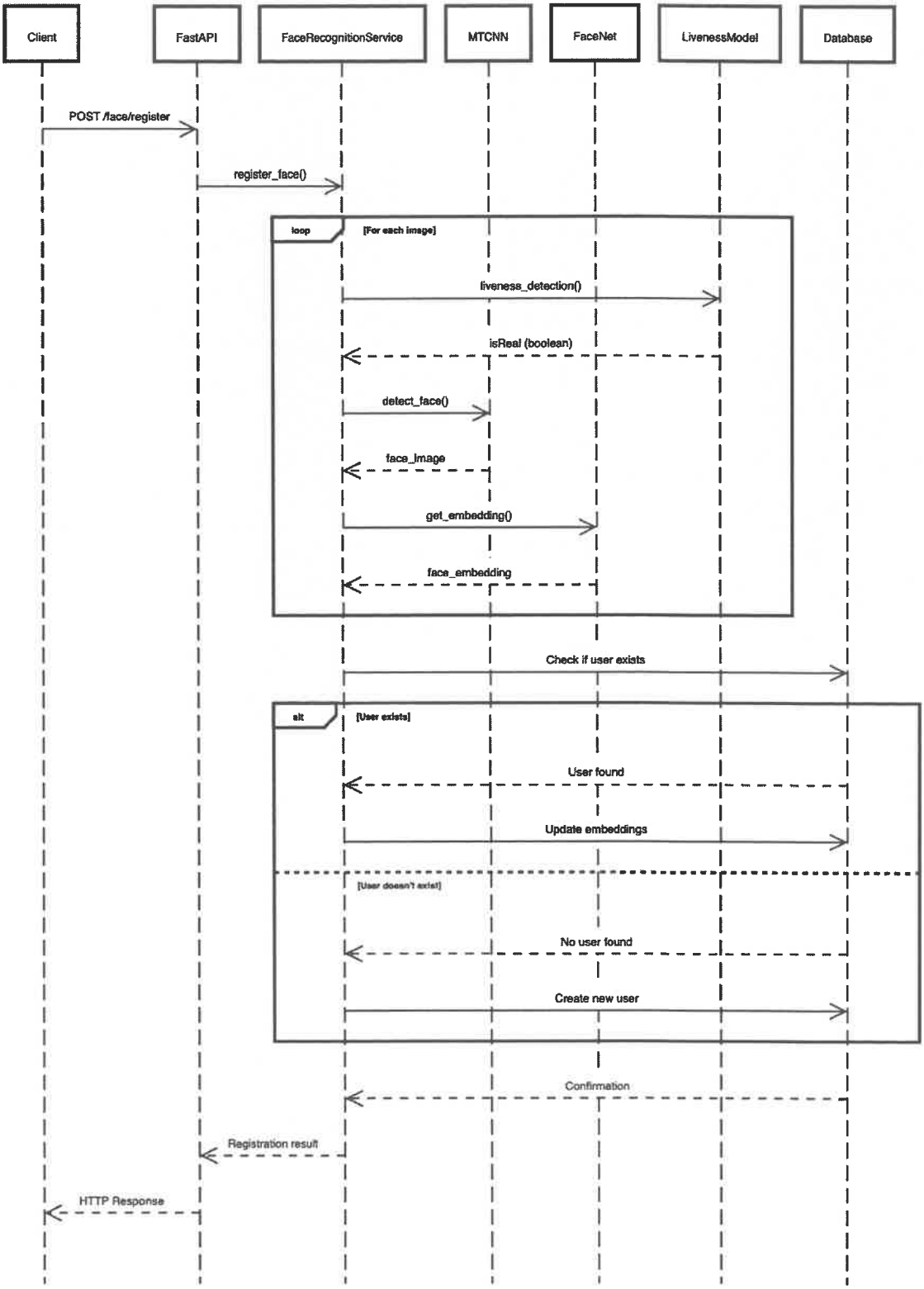
Cấu hình này cho phép API nhận diện khuôn mặt của chúng tôi có thể được truy cập từ internet, mở ra khả năng tích hợp với nhiều ứng dụng khác nhau mà không cần triển khai trên server chuyên dụng.

Tôi đã triển khai các API cho phép các ứng dụng tương tác với hệ thống để xử lý các yêu cầu đăng ký, nhận diện khuôn mặt.

3.2.1. API đăng ký khuôn mặt

Hình 14 mô tả luồng hoạt động của chức năng đăng ký khuôn mặt. Đầu vào của nó là ba ảnh khuôn mặt theo các hướng: nhìn thẳng, quay trái, qua phải và một id tương ứng với một người dùng. Khi người dùng gọi đến API này, các model xử lý khuôn mặt có trong hệ thống như MTCNN, FaceNet và mô hình nhận diện khuôn giả mạo được hệ thống gọi đến để xử lý hình ảnh đầu vào.

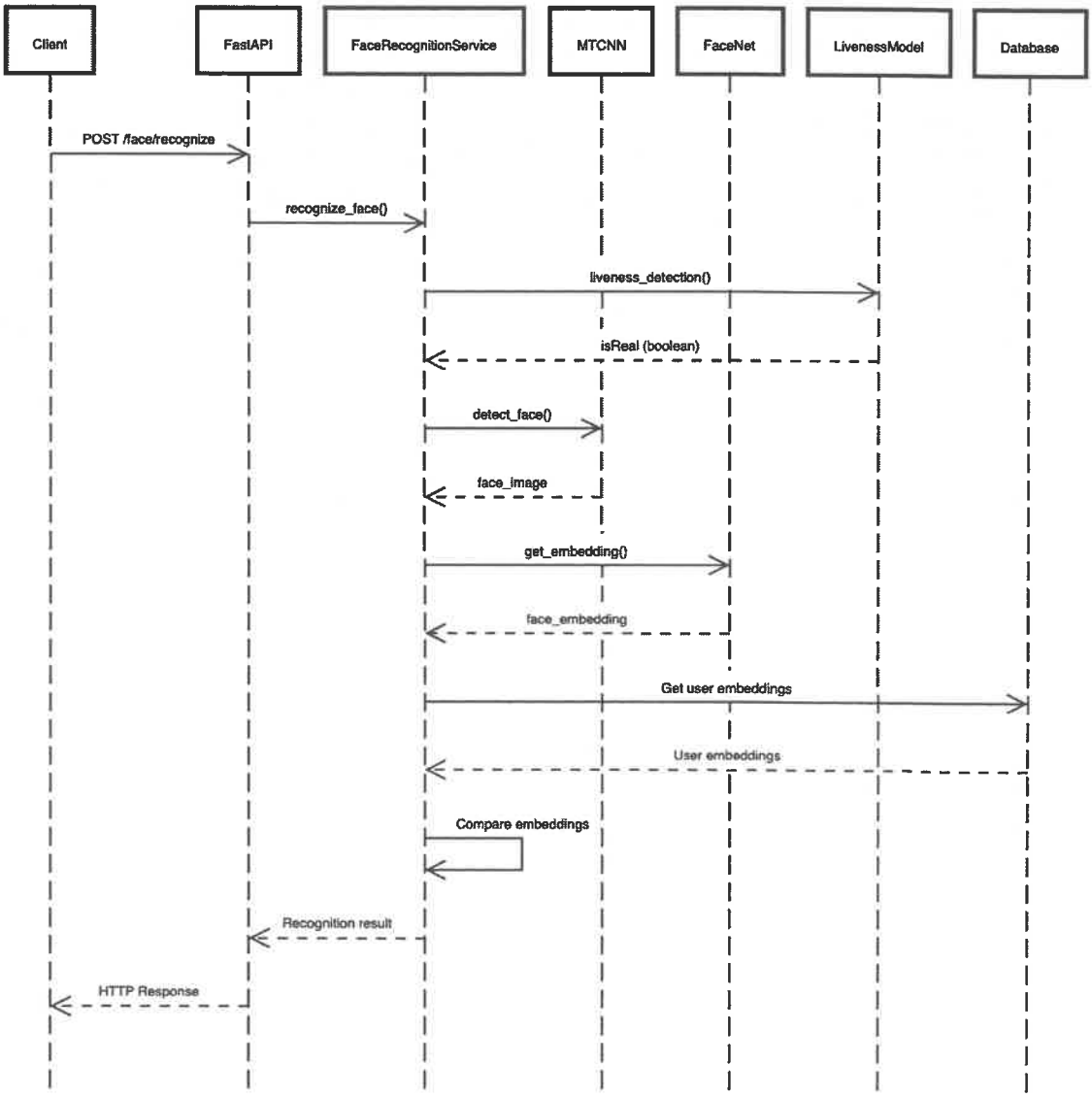
Với mỗi ảnh, đầu tiên hệ thống kiểm tra xem khuôn mặt trong ảnh đó có phải là giả mạo không. Khi đã xác nhận là khuôn mặt thật, MTCNN sẽ được sử dụng để phát xác định được tọa độ khuôn mặt trong ảnh, và trả về ảnh đã cắt theo tọa độ đó. Tiếp đến,hệ thống sẽ trích xuất vector embedding bằng model FaceNet và cập nhật vào cơ sở dữ liệu. Lúc này, hệ thống kiểm tra xem trong cơ sở dữ liệu đã có người dùng nào có id tương ứng với đầu vào hay không. Nếu có, hệ thống sẽ cập nhật các vector embedding cũ bằng các vector embedding vừa được trích xuất. Nếu chưa có, hệ thống tạo mới một đối tượng với id mới và các vector embedding, sau đó cập nhật vào database. Khi quá trình này đã hoàn thành, kết quả sẽ được trả về cho người dùng.



Hình 3.2. Chức năng đăng ký khuôn mặt

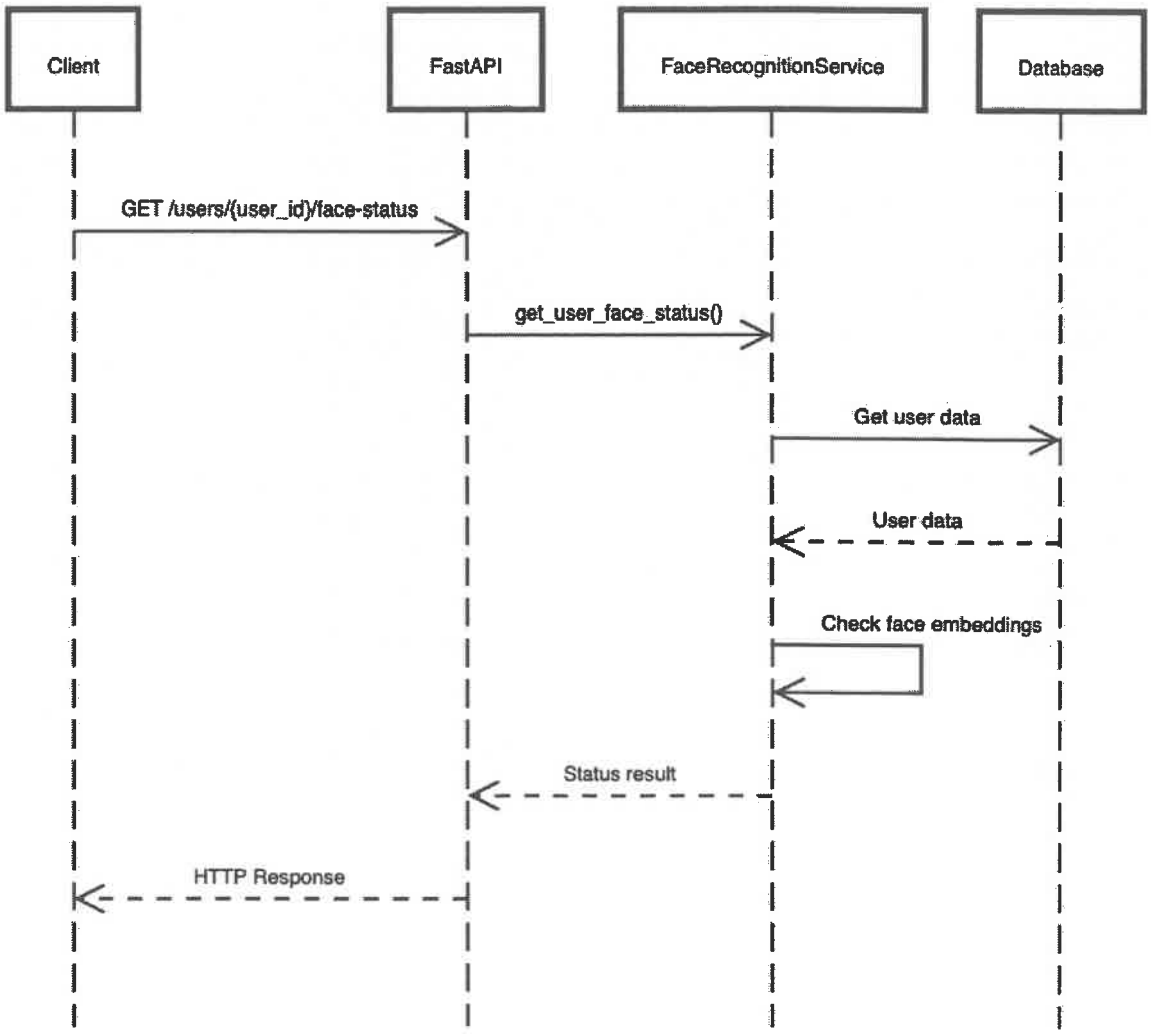
3.2.2. API nhận diện khuôn mặt

API nhận diện khuôn mặt cho phép xác thực danh tính người dùng thông qua khuôn mặt. API này nhận vào ID người dùng và một ảnh khuôn mặt cần nhận diện, sau đó thực hiện quá trình kiểm tra liveness detection để đảm bảo khuôn mặt là thật. Tiếp theo, hệ thống sẽ trích xuất vector đặc trưng (embedding) từ ảnh khuôn mặt và so sánh với các embedding đã lưu trữ trong cơ sở dữ liệu. Kết quả nhận diện được trả về bao gồm thông tin về việc nhận diện thành công hay thất bại, cùng với độ tin cậy của kết quả.



Hình 3.3. Chức năng nhận diện khuôn mặt

3.2.3. API lấy trạng thái bật/tắt nhận diện khuôn mặt

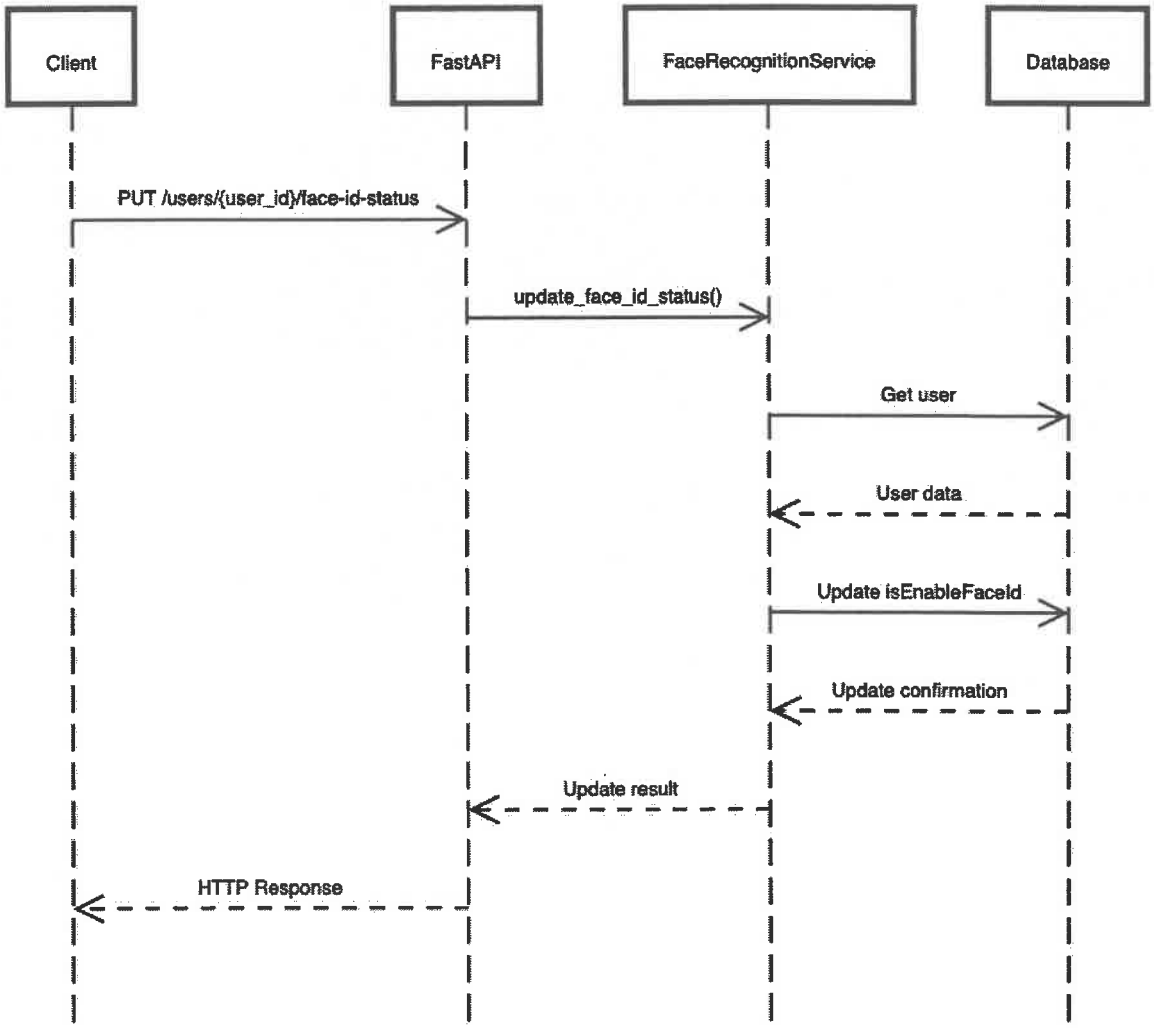


Hình 3.4. Chức năng bật/tắt nhận diện khuôn mặt

3.2.4. API cập nhật/thay đổi trạng thái nhận diện khuôn mặt

API cập nhật/thay đổi trạng thái nhận diện khuôn mặt là một API quan trọng trong hệ thống Face Recognition, cho phép quản lý trạng thái bật/tắt tính năng nhận diện khuôn mặt cho từng người dùng. API này nhận vào ID người dùng và trạng thái mới (bật/tắt) thông qua phương thức PUT, sau đó cập nhật trạng thái trong cơ sở dữ liệu. Khi thành công, API trả về thông báo xác nhận việc cập nhật trạng thái, và trong trường hợp không tìm thấy người dùng, nó sẽ trả về lỗi 404. API này đóng vai trò

quan trọng trong việc giúp các ứng dụng nhận diện khuôn mặt kiểm soát quyền truy cập và quản lý tính năng này trong hệ thống.

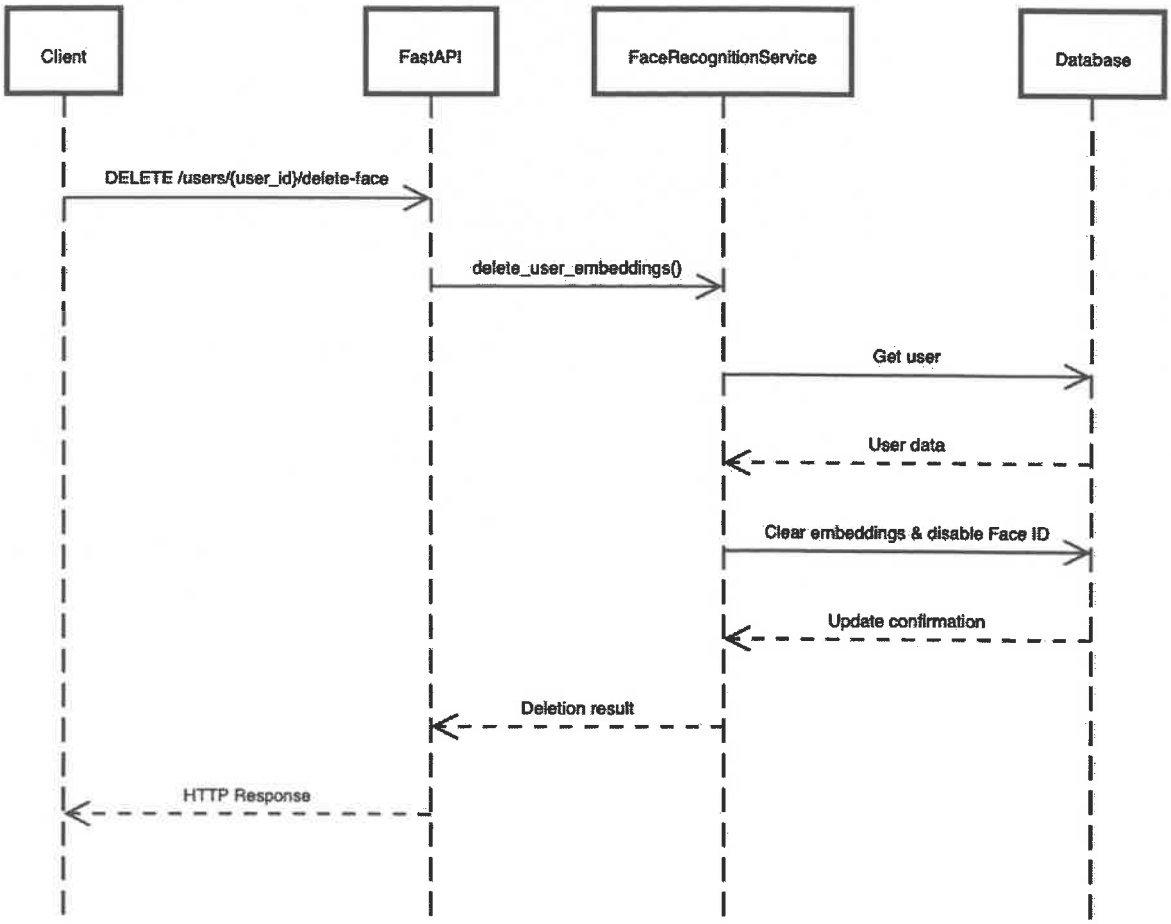


Hình 3.5. Chức năng cập nhật/thay đổi trạng thái nhận diện khuôn mặt

3.2.5. API xóa khuôn mặt đã lưu

API xóa khuôn mặt cho phép xóa toàn bộ dữ liệu khuôn mặt của người dùng khỏi hệ thống. API này nhận vào ID người dùng thông qua phương thức DELETE và thực hiện việc xóa các vector đặc trưng khuôn mặt (embeddings) từ ba góc nhìn (front, left, right) được lưu trữ trong cơ sở dữ liệu. Đồng thời, API cũng tự động tắt tính năng Face ID cho người dùng đó. Khi thực hiện thành công, hệ thống sẽ trả về thông báo xác nhận việc xóa dữ liệu khuôn mặt và tắt Face ID. Trong trường hợp

không tìm thấy người dùng, API sẽ trả về lỗi 404. API này đặc biệt hữu ích trong các trường hợp người dùng muốn xóa dữ liệu khuôn mặt của mình vì lý do bảo mật hoặc khi cần cập nhật lại dữ liệu khuôn mặt mới.



Hình 3.6. Chức năng xóa khuôn mặt đã lưu

3.3. Triển khai trên ứng dụng ONE Home

3.3.1. Ứng dụng ONEHome

Ứng dụng nhà thông minh mà tôi sử dụng trong đề án là ứng dụng ONEHome, một sản phẩm của VNPT-Technology. ONE Home là giải pháp ngôi nhà thông minh cho phép kết nối các thiết bị, giám sát và điều khiển từ xa, tăng cường an toàn, an ninh cho ngôi nhà của bạn.

- Tự động điều khiển các thiết bị theo nhu cầu, thói quen sinh hoạt.

- Cảm nhận môi trường xung quanh và có những phản hồi giúp cuộc sống tiện nghi, thoải mái hơn.
- Dễ dàng điều khiển trên ứng dụng thông qua giọng nói hoặc trực tiếp qua thiết bị.
- Xem trực tiếp với chất lượng Full HD trên thiết bị smart phone mọi nơi mọi lúc.

Tôi đã thêm chức năng nhận diện khuôn mặt trên ứng dụng để tăng cường tính an toàn và bảo mật thông tin. Khi người dùng muốn xem camera trong tài khoản, cần phải xác thực khuôn mặt trước. Nhận diện khuôn mặt còn được áp dụng cho đầu vào của ngữ cảnh tự động. Khi camera phát hiện chuyển động hoặc phát hiện người, nó sẽ đưa ảnh đó vào hệ thống. Nếu hệ thống phát hiện trong ảnh có khuôn mặt của người dùng, nó sẽ kích hoạt các tự động gắn với điều kiện nhận diện khuôn mặt.

3.3.2. Triển khai trên ứng dụng di động OneHome

Trong ứng dụng OneHome, tôi đã thêm những giao diện liên quan đến đăng ký khuôn mặt, xác thực khuôn mặt, cài đặt xác thực khuôn mặt.

Phát hiện khuôn mặt trong ứng dụng ONE Home

Trong đề án này, tôi đã sử dụng ứng dụng ONE Home với điện thoại iPhone. Để có thể gửi ảnh xác thực cho hệ thống, camera của điện thoại phải có thể phát hiện khuôn mặt. Vision Framework là một trong những công nghệ tiên tiến nhất của Apple trong lĩnh vực computer vision, được giới thiệu lần đầu trong iOS 11 và macOS High Sierra. Framework này cung cấp các API mạnh mẽ và dễ sử dụng cho việc phân tích hình ảnh và video, trong đó face detection (nhận dạng khuôn mặt) là một trong những tính năng nổi bật nhất.

Framework này được Apple phát triển để phù hợp với phần cứng thiết bị có trong hệ sinh thái của họ và không được chia sẻ ra ngoài. Framework tự động lựa chọn phần cứng phù hợp nhất để thực thi - từ Neural Engine cho tốc độ cao nhất, đến GPU hoặc CPU tùy theo tình huống. Quá trình xử lý diễn ra hoàn toàn trên thiết bị, đảm bảo tính riêng tư và giảm độ trễ.

Framework cung cấp hai loại request chính cho face detection:

- **VNDetectFaceRectanglesRequest** cho phép phát hiện vị trí các khuôn mặt trong ảnh, trả về bounding boxes với độ chính xác cao và confidence scores.
- **VNDetectFaceLandmarksRequest** đi xa hơn bằng cách nhận dạng các điểm đặc trưng trên khuôn mặt như mắt, mũi, miệng, và đường viền khuôn mặt với độ chi tiết lên đến 76 điểm landmark.

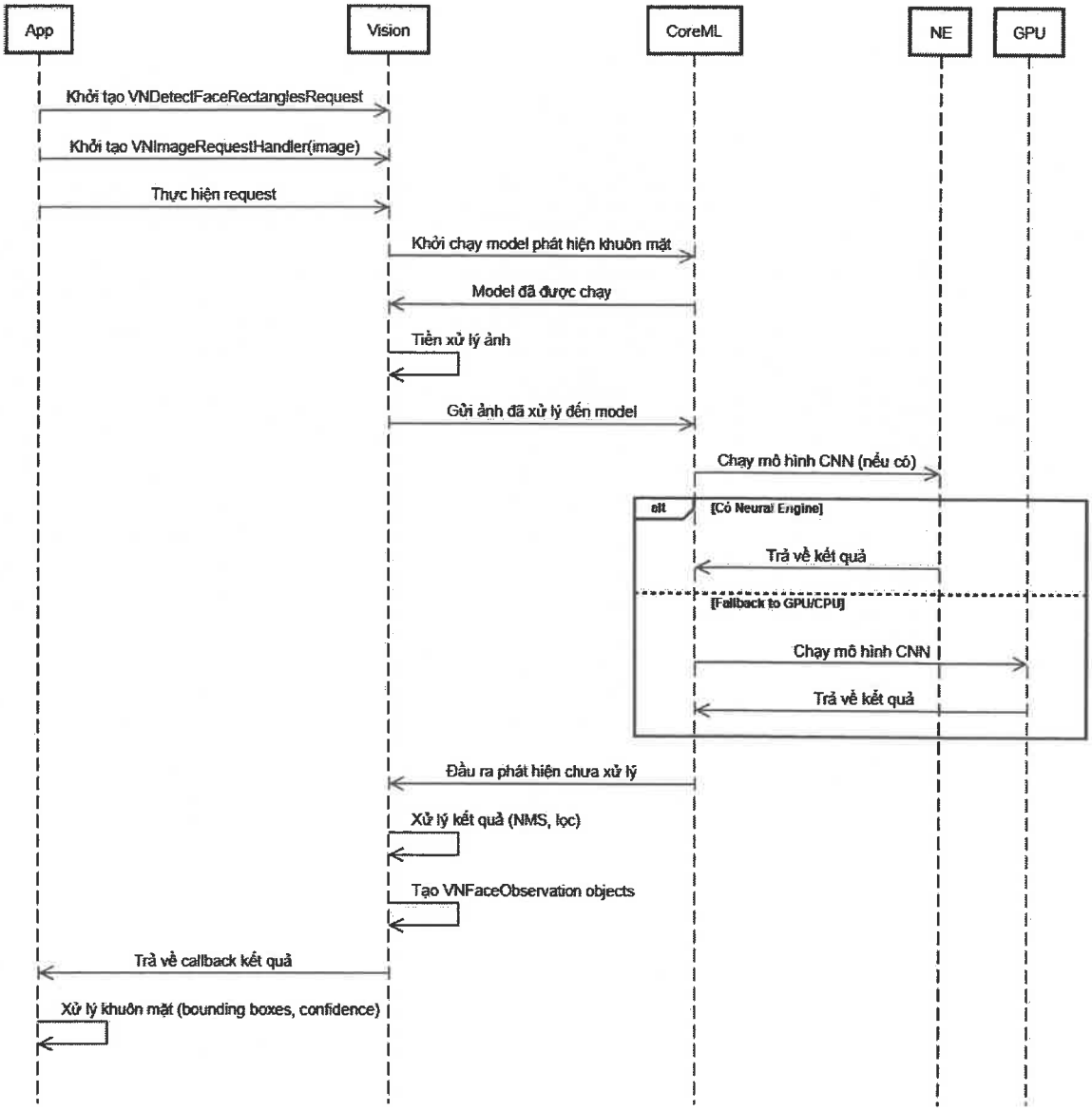
Vision Framework được sử dụng rộng rãi trong nhiều ứng dụng của Apple như Photos app để tự động gắn thẻ khuôn mặt, Camera app cho các tính năng Portrait mode và Face ID.

Sau khi đã có kết quả phát hiện khuôn mặt, lúc này app sẽ tiếp tục kiểm tra tiếp các điều kiện:

- Chỉ có 1 khuôn mặt trong camera
- Toàn bộ khuôn mặt phải đặt trong camera

Sau khi đã thỏa mãn các điều kiện này, app mới gọi các API của hệ thống để thực hiện nhận diện khuôn mặt.

Với hình 3.6, chúng ta có thể thấy được cách tôi đã ứng dụng Vision framework để nhận diện đầu vào khuôn mặt, qua đó lấy được ảnh chứa khuôn mặt cần xử lý cho các bước tiếp theo.



Hình 3.7. Phát hiện khuôn mặt với Vision framework

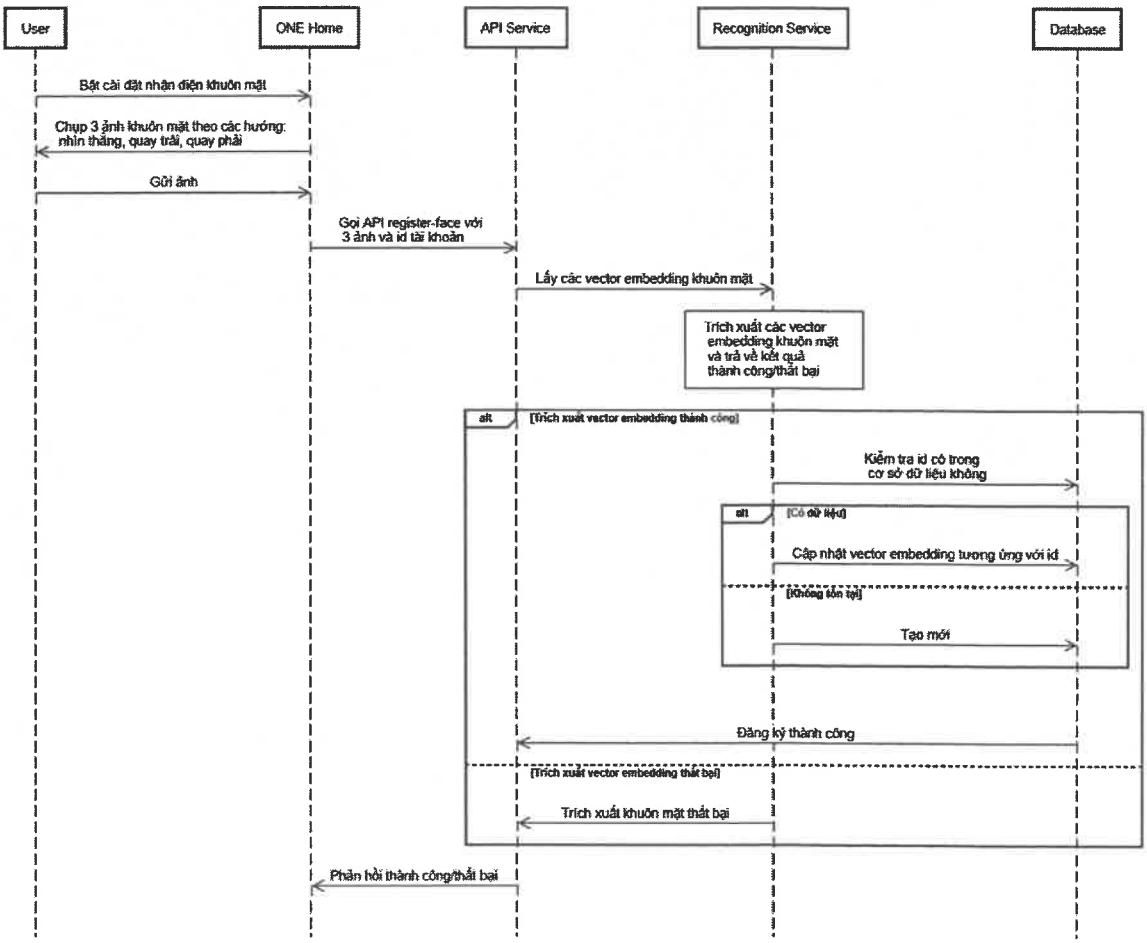
Cài đặt xác thực khuôn mặt

- Mục cài đặt xác thực khuôn mặt nằm trong mục Cài đặt của app, đi từ menu bên trái ứng dụng. Ở đây app sẽ gọi API lấy trạng thái bật/tắt xác thực khuôn mặt của tài khoản. Ngoài ra, chức năng thay đổi khuôn mặt và xóa khuôn mặt sẽ được hiện nếu như trạng thái xác thực khuôn mặt đang được bật.

- Bật cài đặt xác thực khuôn mặt

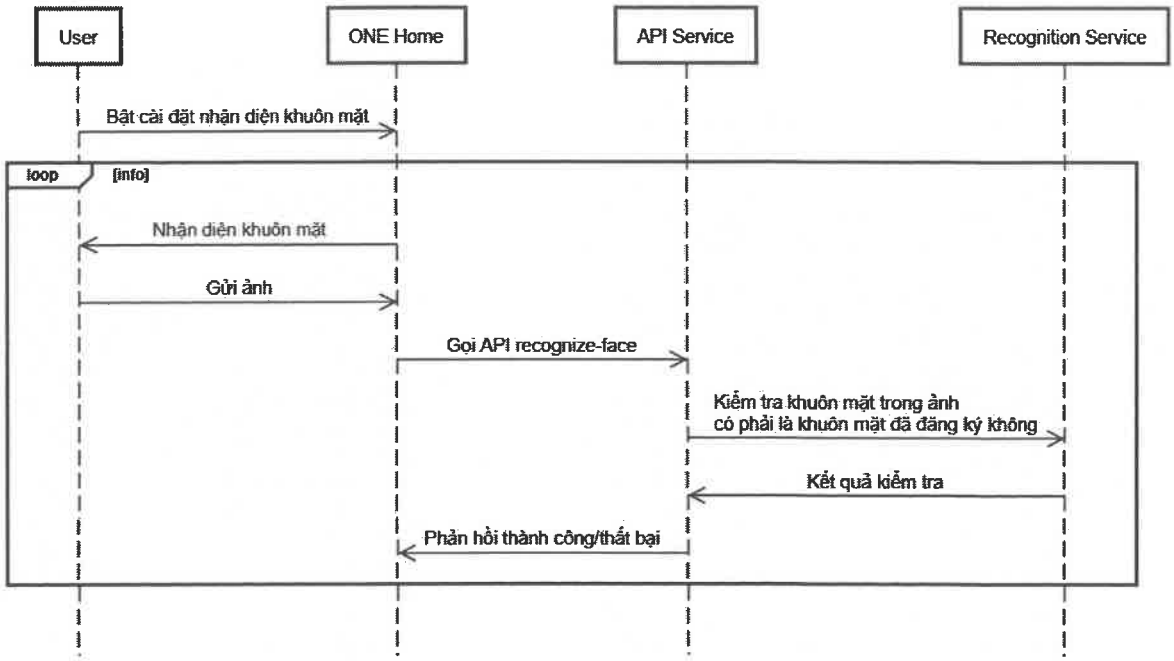
Hình 20 mô tả luồng hoạt động khi bật cài đặt. Trong trường hợp bật cài đặt lần đầu, trong cơ sở dữ liệu chưa có id tài khoản: ứng dụng sẽ mở màn hình thêm

khuôn mặt với lần lượt các hướng dẫn để chụp được 3 ảnh khuôn mặt theo các hướng: nhìn thẳng, quay trái, quay phải. Khi có kết quả, người dùng được điều hướng về lại màn cài đặt kèm với thông báo đăng ký khuôn mặt thành công hoặc thất bại. Còn với trường hợp bật cài đặt sau khi đã xóa dữ liệu khuôn mặt, hệ thống sẽ cập nhật các vector embedding khuôn mặt tương ứng với id tài khoản đó.



Hình 3.8. Luồng hoạt động khi bật cài đặt nhận diện khi chưa có dữ liệu khuôn mặt

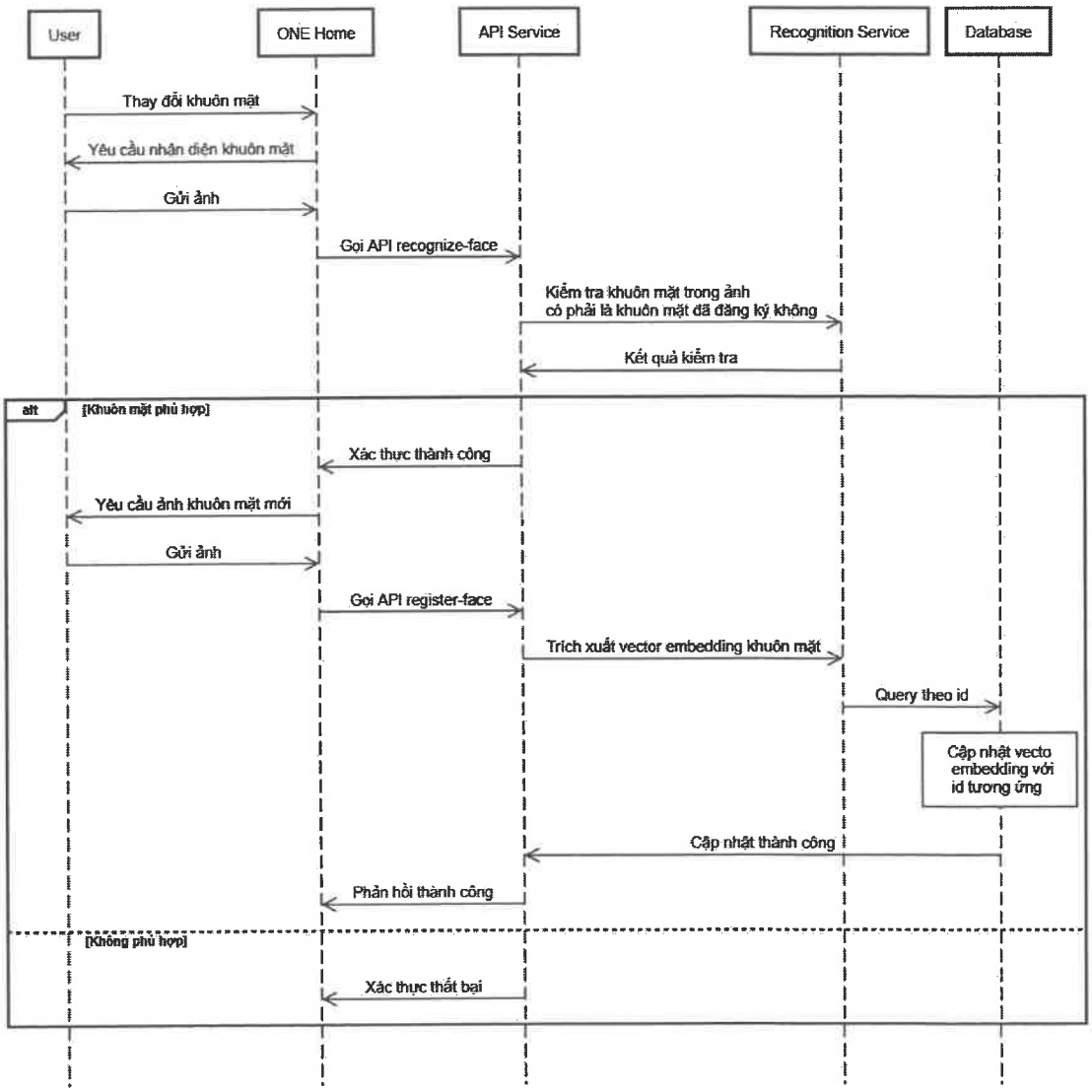
Với trường hợp bật cài đặt, trong cơ sở dữ liệu đã có dữ liệu khuôn mặt: ứng dụng sẽ chuyển hướng đến màn nhận diện khuôn mặt. Sau khi đã xác nhận được ảnh của người dùng khớp với dữ liệu có trong cơ sở dữ liệu thì cài đặt sẽ được bật. Người dùng có tối đa 3 lần để nhận diện.



Hình 3.9. Luồng bật cài đặt khi có dữ liệu khuôn mặt

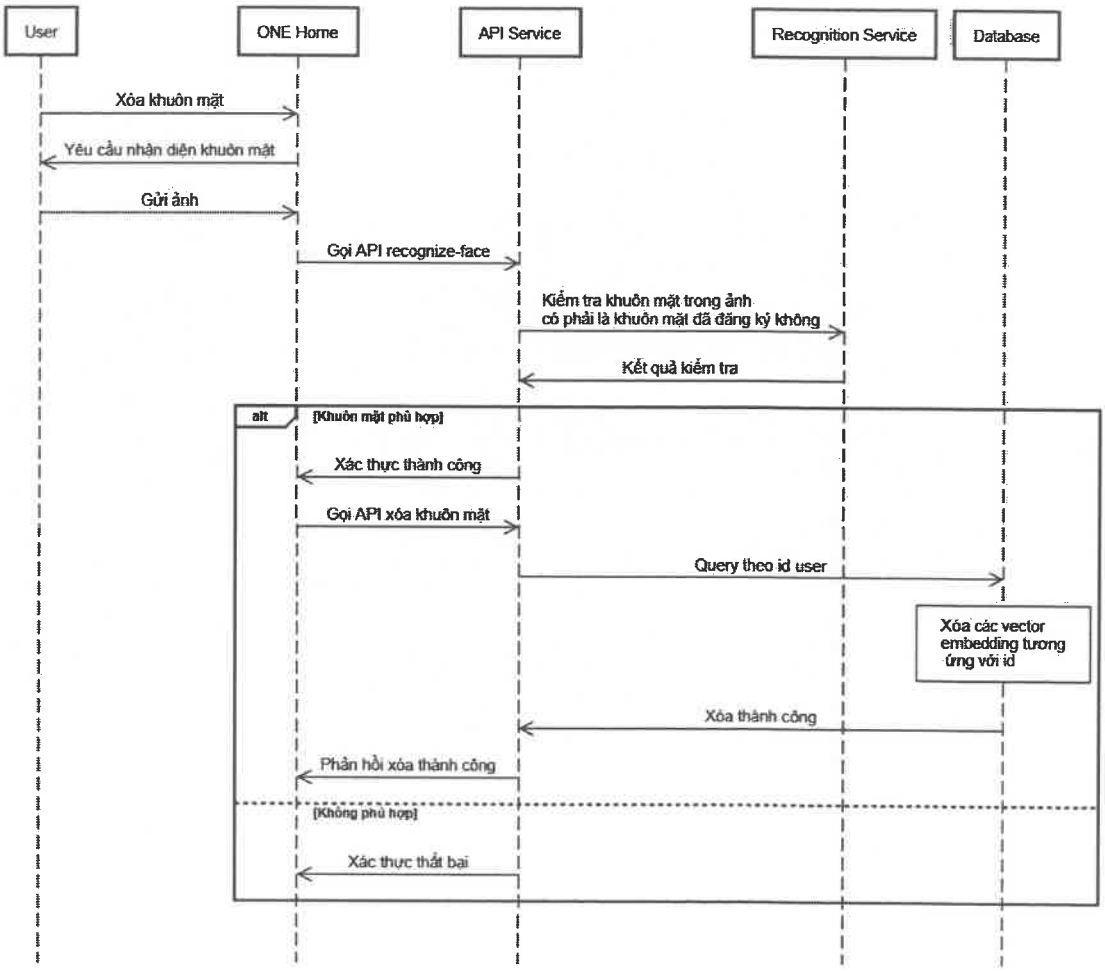
Với những tài khoản lần đầu bật hoặc vừa xóa khuôn mặt xong thì cần phải đăng ký lại khuôn mặt để có thể sử dụng tính năng. Ứng dụng sẽ mở màn hình thêm khuôn mặt với lần lượt các hướng dẫn để chụp được 3 ảnh khuôn mặt theo các hướng: nhìn thẳng, quay trái, quay phải. Khi có kết quả, người dùng được điều hướng về lại màn cài đặt kèm với thông báo đăng ký khuôn mặt thành công hoặc thất bại.

- Để thay đổi khuôn mặt, người dùng cần phải xác thực lại khuôn mặt hiện tại để đảm bảo là chính chủ. Sau khi xác thực thành công thì sẽ điều hướng đến màn đăng ký khuôn mặt để gửi ảnh mới cho hệ thống để tạo dữ liệu khuôn mặt mới thay đổi cho cái cũ.



Hình 3.10. Luồng thay đổi khuôn mặt

- Với xóa khuôn mặt, sau khi đã xác thực thành công, gọi API xóa khuôn mặt tương ứng với tài khoản trong cơ sở dữ liệu. Vì không còn dữ liệu khuôn mặt nữa nên chức năng sẽ tự động tắt đi.



Hình 3.11. Luồng xóa khuôn mặt

Bảo mật cho camera:

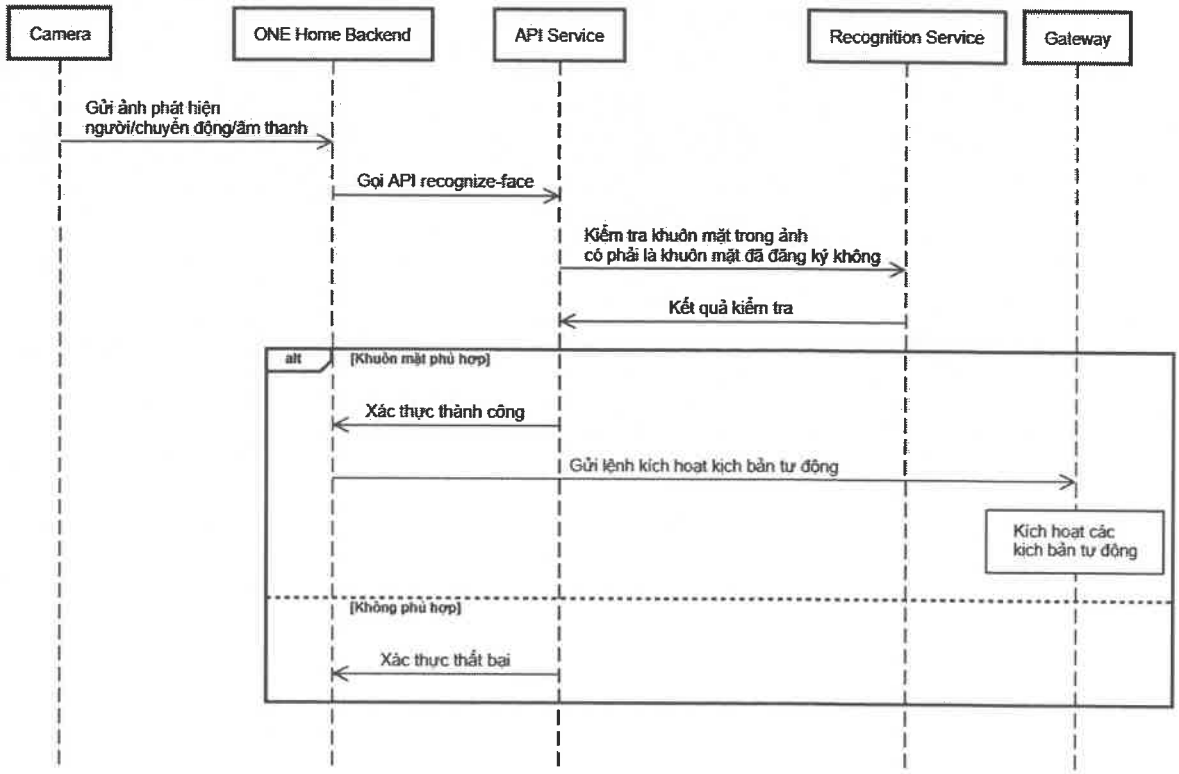
- Để tăng cường bảo mật thông tin cho các dữ liệu nhạy cảm từ camera, ứng dụng sẽ yêu cầu xác thực khuôn mặt với những thiết bị là camera. Để ứng dụng làm được điều này, cần bật tính năng xác thực khuôn mặt trong cài đặt.

- Mỗi khi người dùng bấm vào xem camera từ màn hình chính, từ thông báo, hay từ tìm kiếm, ứng dụng sẽ chuyển hướng đến màn hình yêu cầu xác thực khuôn mặt. Khi đã nhận diện thành công, người dùng mới được chuyển hướng đến màn hình xem camera.

Kích hoạt các kịch bản tự động khi camera nhận diện được khuôn mặt

- Để có thể kích hoạt các tự động dựa trên xác thực khuôn mặt, đầu tiên chức năng xác thực khuôn mặt của ứng dụng phải được bật. Sau đó cứ mỗi khi camera

phát hiện được chuyển động, phát hiện người hoặc phát hiện âm thanh thì camera sẽ chụp lại khoảnh khắc đó. Khi đã có được ảnh, ứng dụng sẽ xác thực với dữ liệu khuôn mặt đã có để xác định xem có người nào trong ảnh phù hợp không. Nếu có thì hệ thống của ONE Home sẽ kích hoạt các tự động tương ứng gắn với điều kiện là phát hiện khuôn mặt của chủ nhà.



Hình 3.12. Luồng hoạt động nhận diện khuôn mặt với ảnh gửi từ camera

3.4. Kết quả thử nghiệm

Hiệu Suất Nhận Diện

Mô hình FaceNet sau khi fine-tuning đạt được hiệu suất cao trong nhận diện khuôn mặt. Trên bộ dữ liệu kiểm thử VGGFace2-MTCNN [5], mô hình đạt:

- Độ chính xác (Accuracy): ~95%
- Độ nhạy (Recall): ~93%
- Độ chuẩn xác (Precision): ~94%
- F1-score: ~93.5%

Kỹ thuật data augmentation đóng vai trò quan trọng trong việc cải thiện hiệu suất, đặc biệt là trong các trường hợp có ít dữ liệu huấn luyện cho mỗi người.

Mô hình liveness được huấn luyện trên bộ dữ liệu CelebA-Spoof [9] với 100,000 hình ảnh, được chia thành 80,000 ảnh cho tập huấn luyện và 20,000 ảnh cho tập kiểm thử. Dữ liệu được xử lý và nạp vào bộ nhớ sử dụng TensorFlow Dataset API với kích thước batch là 128.

Kết quả thực nghiệm trên tập dữ liệu CelebA-Spoof [9] cho thấy mô hình sử dụng MobileNetV2 đạt độ chính xác lên tới 99,4%, vượt nhẹ so với phiên bản sử dụng InceptionNet, đồng thời giảm đáng kể thời gian suy luận (chỉ còn khoảng 35ms mỗi ảnh trên thiết bị phổ thông) và kích thước mô hình (giảm từ 95MB xuống còn 14MB). Điều này chứng minh rằng MobileNetV2 không chỉ phù hợp cho các hệ thống phát hiện giả mạo khuôn mặt thời gian thực mà còn mở ra tiềm năng triển khai rộng rãi trên các thiết bị di động, camera thông minh và nền tảng IoT, nơi tài nguyên tính toán và bộ nhớ bị giới hạn. Như vậy, việc tích hợp MobileNetV2 vào kiến trúc FaceNet là một hướng đi hiệu quả, góp phần nâng cao tính thực tiễn và khả năng ứng dụng của các hệ thống bảo mật sinh trắc học hiện đại.

Hiệu Năng API

API được xây dựng với FastAPI có khả năng xử lý nhiều yêu cầu đồng thời với độ trễ thấp:

- Thời gian phản hồi trung bình của API là 1.2 giây
- Thời gian để phát hiện khuôn mặt là 0.3 giây
- Thời gian để nhận diện khuôn mặt là 0.8 giây
- Thời gian để kiểm tra liveness là 0.4 giây
- Thời gain để đăng ký khuôn mặt là 4-5 giây

Kết luận

Hệ thống đã đáp ứng được các yêu cầu cơ bản về nhận diện khuôn mặt với độ chính xác cao. Các tính năng chính như liveness detection, face registration và

recognition đều hoạt động tốt. Tuy nhiên, vẫn còn một số vấn đề về hiệu suất và bảo mật cần được cải thiện.

3.5. Đánh giá hệ thống

Trong đề án này, chúng tôi đã xây dựng một hệ thống nhận diện khuôn mặt hoàn chỉnh, từ việc tiền xử lý dữ liệu với MTCNN, fine-tuning mô hình FaceNet, phát hiện giả mạo khuôn mặt với MobileNetV2 và xây dựng API với FastAPI và SQLite, đến triển khai dịch vụ với ngrok. Hệ thống đạt được hiệu suất cao trong nhận diện khuôn mặt và có khả năng triển khai trong nhiều ứng dụng thực tế.

Ưu Điểm của Hệ Thống

- Độ chính xác cao: Sử dụng mô hình FaceNet tiên tiến kết hợp với MTCNN cho phép nhận diện khuôn mặt với độ chính xác cao.
- Dễ dàng triển khai: Sử dụng SQLite và FastAPI giúp đơn giản hóa quá trình triển khai.
- Khả năng mở rộng: Kiến trúc của hệ thống cho phép dễ dàng mở rộng và cải tiến.
- Cơ sở dữ liệu SQLite có thể được thay thế bằng các hệ quản trị cơ sở dữ liệu mạnh mẽ hơn như PostgreSQL hoặc MongoDB cho các ứng dụng quy mô lớn.
- Mô hình có thể được tiếp tục fine-tuning với dữ liệu mới mà không cần xây dựng lại toàn bộ hệ thống.
- Truy cập từ xa: Sử dụng ngrok giúp API có thể truy cập từ bất kỳ đâu mà không cần cấu hình phức tạp.

Hạn chế:

- Chưa có các phương pháp bảo mật dữ liệu.
- Tốc độ xử lý thực tế vẫn chưa cao do sự chậm trễ của server ngrok.
- Độ chính xác của model liveness trong thực tế vẫn còn chưa được tốt.
- Chưa có cơ chế cache cho embedding, có thể gây chậm nếu số lượng user tăng lên.
- Chưa có cơ chế load balancing cho việc xử lý nhiều request đồng thời.

CHƯƠNG 4. KẾT LUẬN VÀ KHUYẾN NGHỊ

Trong đề tài này, học viên đã đề xuất và xây dựng mô hình nhận diện khuôn mặt FaceNet với MobileNet V2 làm xương sống, cùng với đó là mô hình nhận diện khuôn mặt giả mạo cũng với MobileNet V2. Các kết quả thực nghiệm cho thấy hệ thống đã chạy và đạt hiệu suất tốt.

Hệ thống là một giải pháp tốt cho việc xác thực danh tính bằng khuôn mặt, với nhiều tính năng bảo mật và độ chính xác cao. Tuy nhiên, vẫn còn một số điểm cần cải thiện để đáp ứng nhu cầu của hệ thống lớn và đảm bảo hiệu suất tối ưu. Với những cải tiến được đề xuất, hệ thống có tiềm năng trở thành một giải pháp xác thực khuôn mặt mạnh mẽ và đáng tin cậy:

- **Cải thiện hiệu suất**: Thử nghiệm với các kiến trúc mạng tiên tiến hơn và kỹ thuật huấn luyện nâng cao.
- **Tối ưu hóa tốc độ**: Áp dụng các kỹ thuật như quantization và pruning để giảm kích thước mô hình và tăng tốc độ suy luận.
- **Bảo mật dữ liệu**: Triển khai các giải pháp mã hóa và bảo mật cho dữ liệu khuôn mặt nhạy cảm.
- **Mở rộng tính năng**: Phát triển thêm các tính năng như nhận diện cảm xúc, ước tính độ tuổi.
- Cần thêm các cơ chế load balancing, caching để tăng hiệu suất, tốc độ phản hồi của hệ thống.

Hệ thống phù hợp cho:

- Hệ thống xác thực trong doanh nghiệp vừa và nhỏ
- Ứng dụng di động cần xác thực khuôn mặt
- Hệ thống kiểm soát ra vào
- Ứng dụng cần xác thực danh tính với độ bảo mật cao

Tuy nhiên, cần lưu ý về khả năng mở rộng và hiệu suất khi triển khai trong môi trường production với số lượng người dùng lớn.

Với hướng nghiên cứu tiếp theo, chúng ta có thể nghiên cứu, phát triển thêm chức năng nhận diện cảm xúc, ước tính độ tuổi. Những chức năng mới này đều mang tính ứng dụng cao trong thực tế. Với nhận diện cảm xúc, chúng ta có thể đánh giá được mức độ hài lòng của khách hàng với sản phẩm. Với ước tính độ tuổi, chúng ta có thể áp dụng vào camera để quản lý trẻ nhỏ.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016), "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks", IEEE Signal Processing Letters.

[2] Schroff, F., Kalenichenko, D., & Philbin, J. (2015), "FaceNet: A Unified Embedding for Face Recognition and Clustering", IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[3] Xinzheng Xu, Meng Du, Huanxiu Guo, Jianying Chang, Xiaoyang Zhao, 2021, "Lightweight FaceNet Based on MobileNet", International Journal of Intelligence Science, 11, 1-16.

[4] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016), "Rethinking the inception architecture for computer vision", IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

[5] Cao, Q., Shen, L., Xie, W., Parkhi, O. M., & Zisserman, A. (2018), "VGGFace2: A dataset for recognising faces across pose and age", IEEE international conference on automatic face & gesture recognition (pp. 67-74).

[6] Zhuchkov, A. (2021), "Analyzing the Effectiveness of Image Augmentations for Face Recognition from Limited Data", International Conference "Nonlinearity, Information and Robotics" (NIR).

[7] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam (2017), "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", Computer Vision and Pattern Recognition.

[8] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen (2018), "MobileNetV2: Inverted Residuals and Linear Bottlenecks", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510-4520.

[9] Yuanhan Zhang, Zhenfei Yin, Yidong Li, Guojun Yin, Junjie Yan, Jing Shao, Ziwei Liu (2020), "CelebA-Spoof: Large-Scale Face Anti-Spoofing Dataset with Rich Annotations", ECCV 2020

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun (2015), "Deep Residual Learning for Image Recognition", IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016

[11] Mingxing Tan, Quoc V. Le (2019), "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", International Conference on Machine Learning (ICML) 2019

[12] Sriram Ram1, Shashaank Vinoth1, Rahul Natesh Gopalakrishnan1, Aastick Amirteswar Balakumar1, Lekshmi Kalinathan1 Thomas Abraham Joseph Velankanni1 (2024), "Leveraging Diverse CNN Architectures for Medical Image Captioning: DenseNet-121, MobileNetV2, and ResNet-50", ImageCLEF 2024

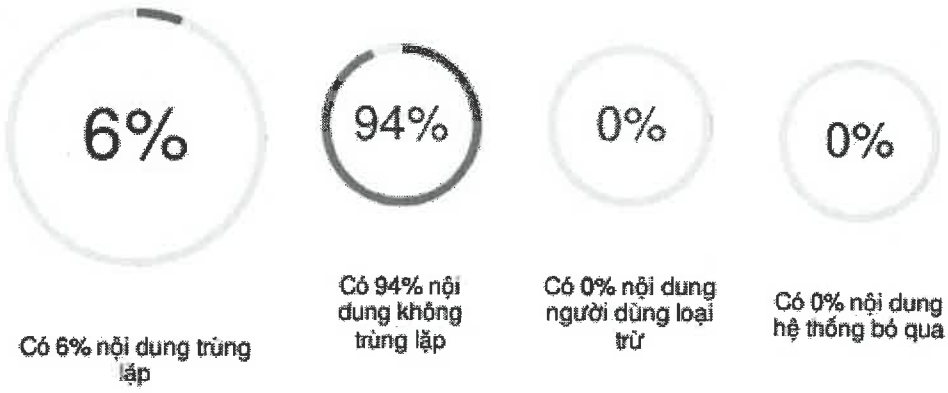
KiểmTraTàiLiệu

BÁO CÁO KIỂM TRA TRÙNG LẶP

Thông tin tài liệu

Tên tài liệu:	NguyenXuanBach_Luan-van-cao-hoc
Tác giả:	Nguyễn Xuân Bách
Điểm trùng lặp:	6
Thời gian tải lên:	23:33 03/08/2025
Thời gian sinh báo cáo:	23:37 03/08/2025
Các trang kiểm tra:	67/67 trang

Kết quả kiểm tra trùng lặp



Nguồn trùng lặp tiêu biểu

tailieu.vn 123docz.net arxiv.org

Học viên
(Ký và ghi rõ họ tên)

Bách

Nguyễn Xuân Bách

Người hướng dẫn khoa học
(Ký và ghi rõ họ tên)

Phương

Vũ Duy Phương

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ
BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BÁO CÁO GIẢI TRÌNH
SỬA CHỮA, HOÀN THIỆN ĐỀ ÁN TỐT NGHIỆP

Họ và tên học viên: Nguyễn Xuân Bách

Chuyên ngành: HTTT

Khóa: 2023 đợt 2

Tên đề tài: Hệ thống xác thực khuôn mặt cho ứng dụng di động

Người hướng dẫn khoa học: TS. Nguyễn Duy Phương

Ngày bảo vệ: 19/07/2025

Các nội dung học viên đã sửa chữa, bổ sung trong đề án tốt nghiệp theo ý kiến đóng góp của Hội đồng chấm đề án tốt nghiệp:

TT	Ý kiến hội đồng	Sửa chữa của học viên
1	Chỉnh sửa lỗi soạn thảo, lỗi ngữ pháp, chính tả	Học viên đã rà soát, chỉnh sửa các lỗi soạn thảo, các lỗi ngữ pháp
2	Chỉnh sửa lại nội dung chương 2	Tiếp thu góp ý của Hội đồng, tác giả đã bổ sung thêm sơ đồ khối kiến trúc của hệ thống, bổ sung thêm mục 2.4.3 Nhận diện khuôn mặt giả mạo
3	Chỉnh sửa lại nội dung chương 3	Tiếp thu góp ý của Hội đồng, tác giả đã bổ sung, phân tích chi tiết hơn về việc xây dựng mô hình nhận dạng khuôn mặt.

Hà Nội, ngày 3 tháng 8 năm 2025

Ký xác nhận của

CHỦ TỊCH HỘI ĐỒNG CHẤM ĐỀ ÁN	THƯ KÝ HỘI ĐỒNG	NGƯỜI HƯỚNG DẪN KHOA HỌC	HỌC VIÊN
PGS.TS.Trần Quang Anh	TS. Đào Thị Thúy Quỳnh	TS. Nguyễn Duy Phương	Nguyễn Xuân Bách

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ
BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BIÊN BẢN
HỌP HỘI ĐỒNG CHẤM ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Căn cứ quyết định số Quyết định số 1098/QĐ-HV ngày 26 tháng 06 năm 2025 của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ. Hội đồng đã họp vào hồi....9....giờ.....25.....phút, ngày 19 tháng 07 năm 2025 tại Học viện Công nghệ Bưu chính Viễn thông để chấm đề án tốt nghiệp thạc sĩ cho:

Học viên: **Nguyễn Xuân Bách**

Tên đề án tốt nghiệp: **Hệ thống xác thực khuôn mặt cho ứng dụng di động**

Chuyên ngành: **Hệ thống thông tin** Mã số: **8480104**

Các thành viên của Hội đồng chấm đề án tốt nghiệp có mặt: ....4.../ 05

TT	HỌ VÀ TÊN	TRÁCH NHIỆM TRONG HĐ	GHI CHÚ
1	**PGS.TS. Trần Quang Anh**	Chủ tịch	
2	**TS. Đào Thị Thúy Quỳnh**	Thư ký	
3	**PGS.TS. Nguyễn Hà Nam**	Phản biện 1	
4	**PGS.TS. Nguyễn Trọng Khánh**	Phản biện 2	
5	**TS. Trần Đăng Công**	Uỷ viên	

Các nội dung thực hiện:

1. Chủ tịch Hội đồng điều khiển buổi họp. Công bố quyết định của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ.
2. Người hướng dẫn khoa học hoặc thư ký đọc lý lịch khoa học và các điều kiện bảo vệ đề án tốt nghiệp của học viên. (có bản lý lịch khoa học và kết quả các môn học cao học của học viên kèm theo).
3. Học viên trình bày tóm tắt đề án tốt nghiệp.
4. Phản biện 1 đọc nhận xét (có văn bản kèm theo)
5. Phản biện 2 đọc nhận xét (có văn bản kèm theo)
6. Các câu hỏi của thành viên Hội đồng:

1) Sự cải tiến Facenet (TLTK số 03) như thế nào?
2) Thời gian nhận dạng có phụ thuộc vào số đối tượng hay không?
3) Đề án đề xuất áp dụng MTCNN cho phát hiện khuôn mặt, phương pháp phát hiện khuôn mặt của Apple? Liệu đề tài có áp dụng được pp đề xuất không?

7. Trả lời của học viên:

.....

- Học viên bổ sung nếu số liệu đối chứng tăng lên thì thời gian thử tăng lên.
- Việc sử dụng thuốc viên nhai dùng không nên là bữa đầu.

8. Thư ký đọc nhận xét về quá trình thực hiện đề án tốt nghiệp của học viên *(có văn bản kèm theo).*

9. Hội đồng họp riêng:

- Bầu Ban kiểm phiếu:
  1. Trưởng Ban kiểm phiếu: TS. Đào Thị Thúy Quỳnh
  2. Uỷ viên Ban kiểm phiếu: PGS.TS. Nguyễn Hà Nam
  3. Uỷ viên Ban kiểm phiếu: TS. Đào Thị Cây
- Hội đồng chấm đề án tốt nghiệp bằng bỏ phiếu kín.
- Ban kiểm phiếu làm việc:
- Trưởng Ban kiểm phiếu báo cáo kết quả kiểm phiếu *(có Biên bản họp Ban kiểm phiếu kèm theo)*
- **Điểm trung bình của đề án tốt nghiệp:** 8,0

Kết luận:

1. Các nội dung cần chỉnh sửa, hoàn thiện sau bảo vệ đề án tốt nghiệp:
   - Chỉnh sửa lại nội dung Chương 2
   - Chỉnh sửa theo ý kiến của Phản biện

2. Đề nghị Học viện công nhận (hoặc không) và cấp bằng (hoặc không) thạc sĩ cho học viên:

3. Đề án tốt nghiệp có thể phát triển thành đề tài nghiên cứu cho NCS.....

Buổi làm việc kết thúc vào..... cùng ngày.

Chủ tịch	**Thư ký**
PGS.TS. Trần Quang Anh	TS. Đào Thị Thúy Quỳnh

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

BẢN NHẬN XÉT LUẬN VĂN TỐT NGHIỆP THẠC SĨ
(Dùng cho người phản biện)

Tên đề tài luận văn: Hệ thống xác thực khuôn mặt cho ứng dụng di động

Chuyên ngành: Hệ thống Thông tin

Mã số: 8.48.01.04

Họ và tên học viên: Nguyễn Xuân Bách

Họ và tên người nhận xét: Nguyễn Trọng Khánh

Học hàm, học vị: PGS. TS.

Cơ quan công tác: Học viên Công nghệ Bưu chính Viễn thông

Số điện thoại : 0912314482 Email: Khanhnt@ptit.edu.vn

NỘI DUNG NHẬN XÉT

I. Cở sở khoa học và thực tiễn, sự cần thiết lựa chọn đề tài:

Đề án nghiên cứu, tìm hiểu một số phương pháp phát hiện, nhận diện khuôn mặt, cũng như phát hiện giả mạo khuôn mặt, sau đó áp dụng các phương pháp trong công nghệ phần mềm đề xây dựng một hệ thống nhận diện khuôn, triển khai trên thiết bị đi động. Mặc dù đây là 1 bài toán đã được đề xuất và triển khai khá nhiều, tuy nhiên gắn liên trong hệ sinh thái sản phẩm phần mềm và phần cứng của các công ty/tập đoàn, thì đây vẫn là 1 bài toán có ý nghĩa thực tiễn trong kinh doanh.

II. Nội dung của luận văn, các kết quả đã đạt được:

Quyển báo cáo đề án được tổ chức khá hợp lý, bao gồm 3 chương, trong đó chương 1 giới thiệu các bước cơ bản, và một số mạng học sâu quan trọng cho phát hiện, nhận diện và chống giả mạo khuôn mặt. Chương 2 đê xuất hệ thống nhận diện khuôn mặt áp dụng các phương pháp đã tìm hiểu ở chương 1. Chương 3 trình bày quá trình triển khai và thử nghiệm hệ thống.

Về cơ bản, đề án đã tìm hiểu được quy trình, cũng như các bước quan trọng trong 1 bài toán nhận diện khuôn mặt và chống giả mảo khuôn mặt. Sau đó đề xuất hệ thống ứng dụng các mô hình trên, trong đó áp dung MTCNN cho phát hiện khuôn mặt, FaceNet (backbone MobinleNet) cho nhận diện khuôn mặt và chống giả mạo khuôn mặt với MobileNetV2. Đề án đã thử nghiệm triển khai và đánh giá các mô hình, cũng như hệ thống nhận diện khuôn mặt trên điện thoại IPhone.

III. Một số nội dung chưa đồng ý

- Nếu xét đây là 1 đề án thiên về khoa học máy tính thì đề án chưa trình bày được các mô hình, hệ thống/ứng dụng liên quan, thay vào đó, đề án giới thiệu trực tiếp luôn các mô hình sẽ áp dụng để xây dựng hệ thống. Còn nếu xét đề án thiên về xây dựng hệ thống, thì báo cáo còn thiếu nội dung liên quan đến thiết kế hệ thống.
- Nội dung liên quan đến thử nghiệm và đánh giá các mô hình chưa thực sự rõ ràng. Đề án mới chỉ huấn luyện và đánh giá việc nhận diên trên bộ dữ liệu có sẵn, còn việc thử nghiệm và đánh giá trên người dùng thực thì chưa có.
- Việc thử nghiệm hiệu năng của các API cũng chưa rõ, API được thử nghiệm trong điều kiện nào, có bao nhiêu người dùng đồng thời, cấu hình máy, mạng như nào …

IV. Những vấn đề học viên cần giải trình thêm:

- Ngoài mạng MTCNN, có một số mạng/phương pháp học máy cổ điển cũng thường được áp dụng để phát hiện khuông mặt, học viên có biết những/mạng phương pháp đó không? Nếu biết, tại sao lại dùng mạng MTCNN để phát hiện khuôn mặt, trong khi có nhiều mạng học máy cổ điển khác cũng cho phép phát hiện khuôn mặt với tốc độ nhanh hơn, hiệu năng giảm không quá nhiều.
- Đề án đề xuất áp dụng MTCNN cho phát hiện khuôn mặt, nhưng khi triển khai với OneHome lại sử dụng thư viện nhận diện khuôn mặt của nền tảng Apple ? Liệu đề án có triển khai đúng phương pháp đề xuất không?

V. Kết luận

Đồng ý cho phép học viên bảo vệ luận văn tốt nghiệp.

Hà Nội, Ngày 15 tháng 7.. năm 2025

NGƯỜI NHẬN XÉT

(Ký và ghi rõ họ tên)

Nguyễn Trọng Khánh

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BẢN NHẬN XÉT LUẬN VĂN TỐT NGHIỆP THẠC SĨ
(*Dùng cho người phản biện*)

Tên đề tài luận văn: Hệ thống xác thực khuôn mặt cho ứng dụng di động
Chuyên ngành: Khoa học máy tính Mã số:8.48.01.01
Tên học viên: Nguyễn Xuân Bách
Họ và tên người nhận xét: Nguyễn Hà Nam.....
Học hàm, học vị: PGS. TS Chuyên ngành: CNTT .....
Cơ quan công tác: Ban Khoa học và Đổi mới sáng tạo, ĐHQGHN.....

NỘI DUNG NHẬN XÉT

I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài:
Nhận diên khuôn mặt là một bài toán cơ bản trong thị giác máy tính với nhiều ứng dụng trong nhiều sản phẩm quan trọng khác nhau và ngày càng trở thành một công nghệ rất cần thiết trong lĩnh vực CNTT. Nội dung luận văn là tìm hiểu phương pháp hiệu quả để nhận dạng khuôn mặt trong các hệ thống không đủ các tài nguyên như trong điện thoại di động là một hương nghiên cứu có tính thời sự và thực tiễn cao.
II/ Về nội dung, chất lượng của luận văn, các kết quả đã đạt được (so với đề cương đã được duyệt):
Nội dung của luận văn và các kết quả đạt được cơ bản bám sát theo đề cương đã được phê duyệt. Luận văn được trình bày trong 3 chương ngoài phần giới thiệu và kết luận với các nội dung chính sau: giới thiệu các kỹ thuật học sâu phù hợp với bài toán nhận diện khuôn mặt; Phân tích và thiết kế hệ thống và triển khai thử nghiệm. Về cơ bản hệ thống đã được xây dựng và thử nghiệm trên bộ dữ liệu thực tế và cho kết quả bước đầu.
III/ Những vấn đề cần giải thích thêm:
- Giải thích sự cải tiến học viên so với kiến trúc FaceNet trong tài liệu tham khảo số 3?
- Phần đề xuất của học viên (1.3) nên được chuyển sang chương 2 với các phân tích chi tiết hơn.
- Chương 2 cần bổ sung sơ đồ khối của hệ thống. Mục 1.3 nên ghép với 3.1 và 3. Để thành nội dung nghiên cứu xây dựng mô hình nhận dạng khuôn mặt. Phần phân tích thiết kế hệ thống nên tách thành 1 chương riêng.
- Còn nhiều lỗi chính tả cần được rà soát và chỉnh sửa.

IV/ Kết luận:
Luận văn đáp ứng được yêu cầu cơ bản của luận văn thạc sỹ chuyên ngành KHMT(theo định hướng ứng dụng). Tôi đồng ý để học viên được bảo vệ luận văn trước Hội đồng chấm luận văn thạc sĩ

Ngày 15 tháng 7 năm 2025
NGƯỜI NHẬN XÉT
(Ký và ghi rõ họ tên)

Nguyễn Hà Nam