ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN-ĐIỆN TỬ

BỘ MÔN VIỄN THÔNG

----------o0o----------

BÁO CÁO ĐỒ ÁN 2

HỆ THỐNG MỞ KHÓA CỬA THÔNG MINH BẰNG NHẬN DIỆN KHUÔN MẶT

GVHD: Ths. Nguyễn Khánh Lợi

Sinh viên thực hiện: Lâm Gia Bảo – 2210214

Thành phố Hồ Chí Minh, 01/2026

LỜI CẢM ƠN

Em xin gửi lời cảm ơn sâu sắc đến thầy Nguyễn Khánh Lợi vì đã luôn hỗ trợ em ngay từ những ngày đầu nêu lên ý tưởng và thực hiện đề tài. Sự đồng hành và hướng dẫn tận tình của thầy đã giúp em có được định hướng rõ ràng và điều kiện thuận lợi nhất để hoàn thành đề tài này. Em xin kính chúc thầy luôn dồi dào sức khỏe, hạnh phúc và tràn đầy niềm vui trong cuộc sống.

Hồ Chí Minh, ngày 5 tháng 1 năm 2026

Lâm Gia Bảo

TÓM TẮT ĐỀ TÀI

Đề tài đồ án 2 của em tập trung xây dựng một Hệ thống khóa cửa thông minh này là một giải pháp kiểm soát ra vào hiện đại, tích hợp công nghệ nhận diện khuôn mặt và nền tảng IoT đám mây. Quy trình hoạt động bắt đầu khi camera quét và xác thực khuôn mặt, sau đó tự động ghi lại nhật ký truy cập (bao gồm tên và thời gian) lên Firebase. Một ứng dụng di động, được xây dựng trên nền tảng Thunkable, sẽ liên tục đồng bộ và truy xuất dữ liệu mới nhất từ Firebase để đối chiếu với cơ sở dữ liệu người dùng đã đăng ký. Trong trường hợp xác thực thành công, người dùng sẽ được cấp quyền sử dụng chức năng "Mở khóa" trên ứng dụng. Khi kích hoạt, ứng dụng sẽ truyền tín hiệu điều khiển không dây đến vi điều khiển ESP32 để thực thi lệnh mở cửa. Ngược lại, nếu khuôn mặt không hợp lệ, chức năng mở khóa trên ứng dụng sẽ bị vô hiệu hóa, đảm bảo an ninh. Giải pháp này cung cấp cơ chế xác thực hai lớp (sinh trắc học và ứng dụng), mang lại sự an toàn, tiện lợi và khả năng giám sát lịch sử ra vào từ xa. Trong đề tài này nội dung có 3 phần chính:

1) TỔNG QUAN HỆ THỐNG
2) TỔNG QUAN LÝ THUYẾT
3) KẾT QUẢ THIẾT KẾ HỆ THỐNG NHẬN DIỆN

MỤC LỤC

TÓM TẮT ĐỀ TÀI 2

I. TỔNG QUAN HỆ THỐNG 6

1. Các linh kiện sử dụng 6
2. Sơ đồ nguyên lý 8
3. Sơ đồ khối hệ thống 8
4. Quy trình đóng/mở khóa 11
5. Quy trình đăng nhập 12
6. Quy trình đăng ký 13

II. TỔNG QUAN LÝ THUYẾT 14

1. Tìm hiểu về ngôn ngữ lập trình python với AI 14
   - 1.1 Giới thiệu Python 14
   - 1.2 Python và trí tuệ nhân tạo (AI) 16
2. Tìm hiểu về Firebase 17
   - 2.1 Giới thiệu Firebase 17
   - 2.2 Cách hoạt động của Firebase 20
3. Tìm hiểu về Thunkable 20
4. Nhận diện khuôn mặt 21
   - 4.1 Mô hình MTCNN (Multi-task Cascaded Convolutional Networks) 22
   - 4.2 Mô hình FaceNet 24
5. Cơ chế giao tiếp HTTP giữa Thunkable và ESP32 29
   - 5.1 Cấu trúc của HTTP Request 29
   - 5.2 Các phương thức chính trong phương pháp HTTP Request 30

III. KẾT QUẢ THIẾT KẾ HỆ THỐNG NHẬN DIỆN 32

1. Đóng gói ứng dụng Backend (Containerization) 32

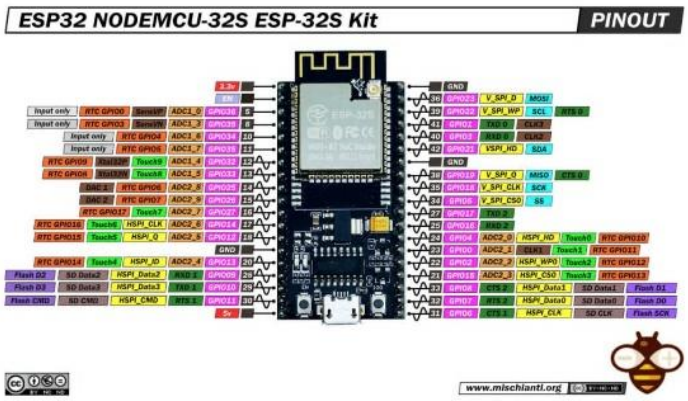
2. Triển khai hệ thống nhận diện trên Google Cloud 32
3. App Mobile 33
3.1 Giao diện chính 33
3.2 Giao diện đăng ký 34
3.3 Giao diện đóng/mở khóa 35
4. Cơ sở dữ liệu (database) 36
TÀI LIỆU THAM KHẢO 38

DANH MỤC HÌNH ẢNH

Hình 1 Sơ đồ chân ESP32 NodeMCU-32S 6
Hình 2 Khóa chốt điện solenoid 12Vdc LY-03 6
Hình 3 Module relay 5V 7
Hình 4 Adaptor AC-DC 12V 2A 7
Hình 5 Sơ đồ nguyên lý 8
Hình 6 Các tính năng của Firebase 18
Hình 7 Kiến trúc mạng P-Net 22
Hình 8 Kiến trúc mạng R-Net 22
Hình 9 Kiến trúc mạng O-Net 23
Hình 10 Kiến trúc mô hình FaceNet 25
Hình 11 Phép tích chập 26
Hình 12 Quá trình tích chập với 3×3 filter 26
Hình 13 Feature detector 26
Hình 14 Valid padding, Same padding và Full padding 27
Hình 15 Max Pooling 28
Hình 16 Các thành phần trong cấu trúc của Http Request 29
Hình 17 Ví dụ về Request header giúp gửi các yêu cầu từ client đến server 30
Hình 18 Các phương thức trong HTTP Request 31
Hình 19 Quá trình đóng gói ứng dụng và thiết lập môi trường thực thi với Docker 32
Hình 20 Quá trình Deploy Container Image từ Google Container Registry (GCR) lên môi trường Serverless 32
Hình 21 Giao diện chính 33
Hình 22 Giao diện đăng ký 34
Hình 23 Giao diện đóng/mở khóa 35
Hình 24 Trạng thái cơ sở dữ liệu trước khi có truy cập 36
Hình 25 Kết quả ghi log sau khi người dùng đăng nhập thành công 36
Hình 26 Kho lưu trữ dữ liệu khuôn mặt trước khi đăng ký 37
Hình 27 Dữ liệu Face Embedding được sinh ra sau khi đăng ký thành công 37

I. TỔNG QUAN HỆ THỐNG

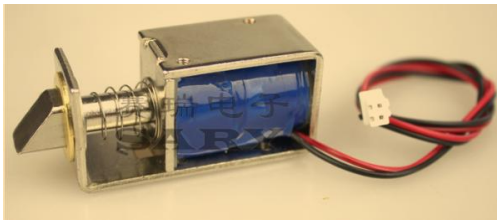
1. Các linh kiện sử dụng



Hình 1 Sơ đồ chân ESP32 NodeMCU-32S

Kit Wifi BLE ESP32 NodeMCU-32S CH340 của Ai-Thinker là giải pháp phát triển IoT toàn diện, được xây dựng dựa trên vi điều khiển trung tâm ESP32 SoC mạnh mẽ. Kế thừa ưu điểm từ dòng NodeMCU ESP8266, sản phẩm mang lại trải nghiệm sử dụng dễ dàng với thiết kế tích hợp sẵn mạch nạp và giao tiếp UART CH340, hỗ trợ song song kết nối Wifi chuẩn 802.11 b/g/n/e/i và Bluetooth BLE 4.2. Đây là lựa chọn tối ưu cho các kỹ sư và nhà nghiên cứu trong việc triển khai các ứng dụng điều khiển từ xa, thu thập dữ liệu và xây dựng mạng lưới vạn vật kết nối (IoT).

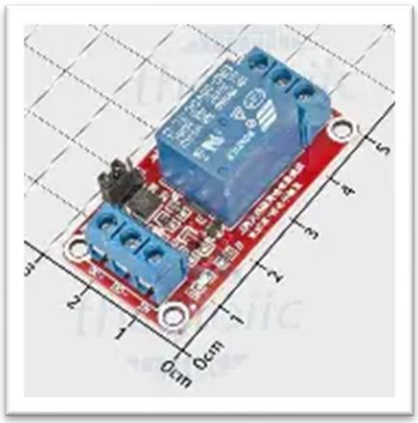
Về mặt phần cứng, bo mạch sở hữu cấu hình ấn tượng với bộ nhớ SPI Flash 32Mbits và dải tần hoạt động rộng từ 2400~2483.5Mhz. Kit được thiết kế theo chuẩn 38 chân cắm (khoảng cách 2.54mm), cho phép truy xuất đầy đủ các giao tiếp ngoại vi đa dạng như UART, SPI, I2C, PWM, ADC và DAC. Với kích thước nhỏ gọn (25.4 x 48.3mm), cấp nguồn tiện lợi qua cổng Micro USB 5V cùng các nút chức năng tích hợp sẵn (BOOT, ENABLE), thiết bị đảm bảo tính linh hoạt và độ ổn định cao trong quá trình vận hành hệ thống.



Hình 2 Khóa chốt điện solenoid 12Vdc LY-03

Khóa chốt điện Solenoid Lock LY-03 là thiết bị an ninh hoạt động dựa trên nguyên lý điện từ (Solenoid), được thiết kế kèm gá chốt đồng bộ để hoạt động như một ổ khóa cửa tự động. Với cấu trúc thường đóng (Normally Closed) mang lại độ an toàn cao, thiết bị này là thành phần quan trọng trong các hệ thống nhà thông minh, tủ locker điện tử và các loại cửa đóng mở bằng điện. Sản phẩm nổi bật nhờ độ bền cơ học tốt, chất lượng hoàn thiện chắc chắn, đáp ứng hiệu quả nhu cầu tự động hóa và bảo mật hiện đại.

Thiết bị cung cấp tùy chọn điện áp linh hoạt 12VDC hoặc 24VDC, vận hành với dòng điện 0.8A và công suất tiêu thụ 9.6W. Khóa sở hữu tốc độ phản ứng cực nhanh (dưới 1 giây), đảm bảo thao tác đóng mở tức thời. Tuy nhiên, người dùng cần đặc biệt lưu ý giới hạn thời gian kích điện liên tục phải dưới 10 giây để bảo vệ cuộn hút solenoid khỏi quá nhiệt và đảm bảo tuổi thọ lâu dài cho thiết bị.

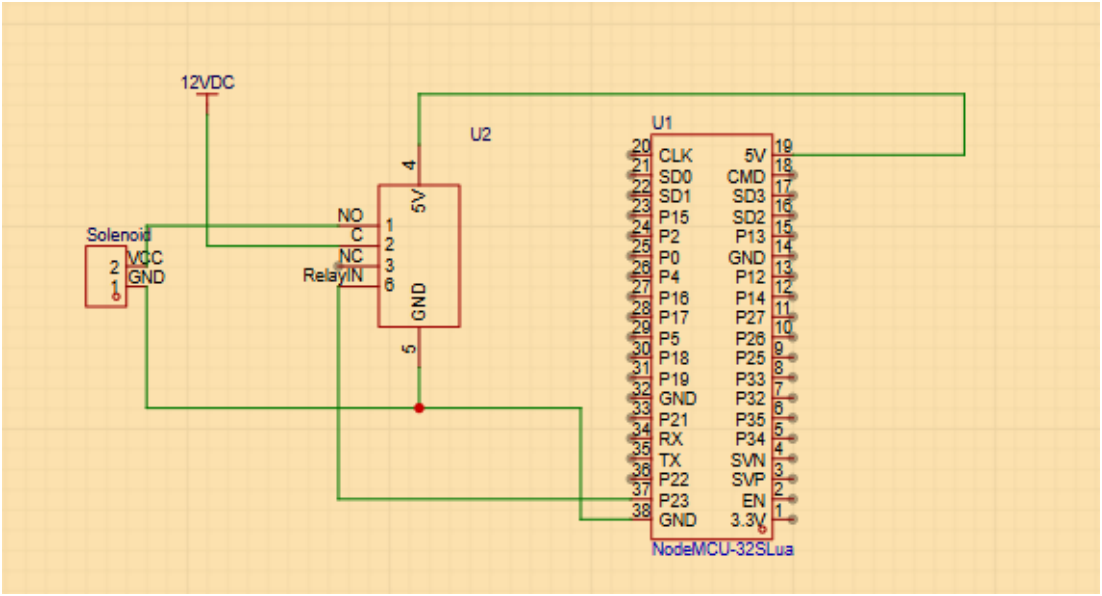


Hình 3 Module relay 5V

Hình 4 Adaptor AC-DC 12V 2A

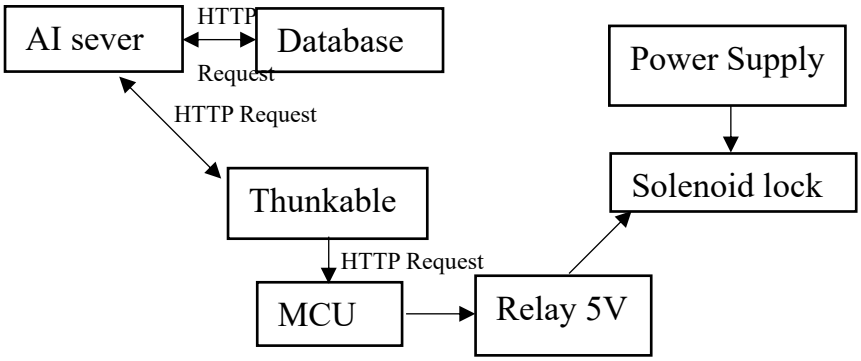
Bộ chuyển đổi nguồn AC-DC sử dụng công nghệ nguồn xung ổn định, hỗ trợ dải điện áp đầu vào rộng 100~240VAC (50/60Hz) và cung cấp đầu ra chuẩn 12VDC với dòng tải tối đa 2A. Để đảm bảo hiệu suất và tuổi thọ linh kiện, thiết bị được khuyến nghị vận hành ở mức 70% công suất đối với các tác vụ liên tục. Sản phẩm được thiết kế với dây dẫn dài 150cm kết thúc bằng giắc DC tròn kích thước 5.5x2.1~2.5mm, đảm bảo khả năng kết nối chắc chắn và tương thích cao với đa số các thiết bị điện tử tiêu chuẩn hiện nay.

2. Sơ đồ nguyên lý



Hình 5 Sơ đồ nguyên lý

3. Sơ đồ khối hệ thống



- Khối Giao diện & Điều khiển (Thunkable)

Khối này đóng vai trò là cổng tương tác trực tiếp giữa người dùng và hệ thống, được xây dựng trên nền tảng Thunkable. Chức năng chính của khối là thu thập dữ liệu đầu vào thông qua việc chụp ảnh khuôn mặt người dùng và gửi yêu cầu xử lý đến máy chủ đám mây. Ứng dụng cũng đảm nhiệm vai trò điều phối luồng logic: sau khi nhận kết quả định danh từ Google Cloud, nếu dữ liệu trùng khớp với người dùng đã đăng ký, ứng dụng sẽ tự động kích hoạt giao diện điều khiển và gửi tín hiệu mở khóa thông qua giao thức HTTP Request đến địa chỉ IP của khối điều khiển phần cứng.

- Khối AI Sever

Đây là trung tâm xử lý trí tuệ nhân tạo của hệ thống, được triển khai trên nền tảng Google Cloud (Cloud Run/Functions). Khối này tiếp nhận các đường dẫn ảnh (URL) từ ứng dụng di động, thực hiện các thuật toán Tiền xử lý (Preprocessing) và Trích xuất đặc trưng (Feature Extraction) để phân tích khuôn mặt. Dựa trên kết quả phân tích, hệ thống sẽ thực thi logic rẽ nhánh: đối với khuôn mặt mới, hệ thống tiến hành quy trình đăng ký và mã hóa dữ liệu; đối với khuôn mặt đã tồn tại, hệ thống thực hiện so khớp và trả về kết quả định danh (tên người dùng) cho ứng dụng di động để quyết định quyền truy cập.

- Khối Database

Khối này sử dụng dịch vụ Firebase (Firestore và Storage) để đảm bảo tính toàn vẹn và khả năng truy xuất dữ liệu nhanh chóng. Nhiệm vụ cốt lõi của khối là lưu trữ an toàn các vector đặc trưng khuôn mặt (Face Embeddings) và thông tin định danh người dùng phục vụ cho quá trình so khớp. Bên cạnh đó, khối này còn đóng vai trò quan trọng trong việc quản lý an ninh thông qua chức năng ghi nhật ký hệ thống (System Logs), tự động lưu lại lịch sử truy cập, thời gian và danh tính người dùng mỗi khi có yêu cầu đăng nhập từ ứng dụng Thunkable.

- Khối MCU

Khối này sử dụng vi điều khiển ESP32 làm hạt nhân xử lý, đóng vai trò là một IoT Node kết nối hạ tầng mạng với các thiết bị vật lý. ESP32 được lập trình để vận hành như một Web Server đơn giản, liên tục lắng nghe các yêu cầu

mạng. Khi nhận được tín hiệu điều khiển HTTP Request hợp lệ từ ứng dụng di động, vi điều khiển sẽ xử lý và xuất tín hiệu số (Digital Signal) ra các chân GPIO tương ứng để kích hoạt khối chấp hành, đồng thời có thể phản hồi trạng thái hoạt động ngược lại cho người dùng.

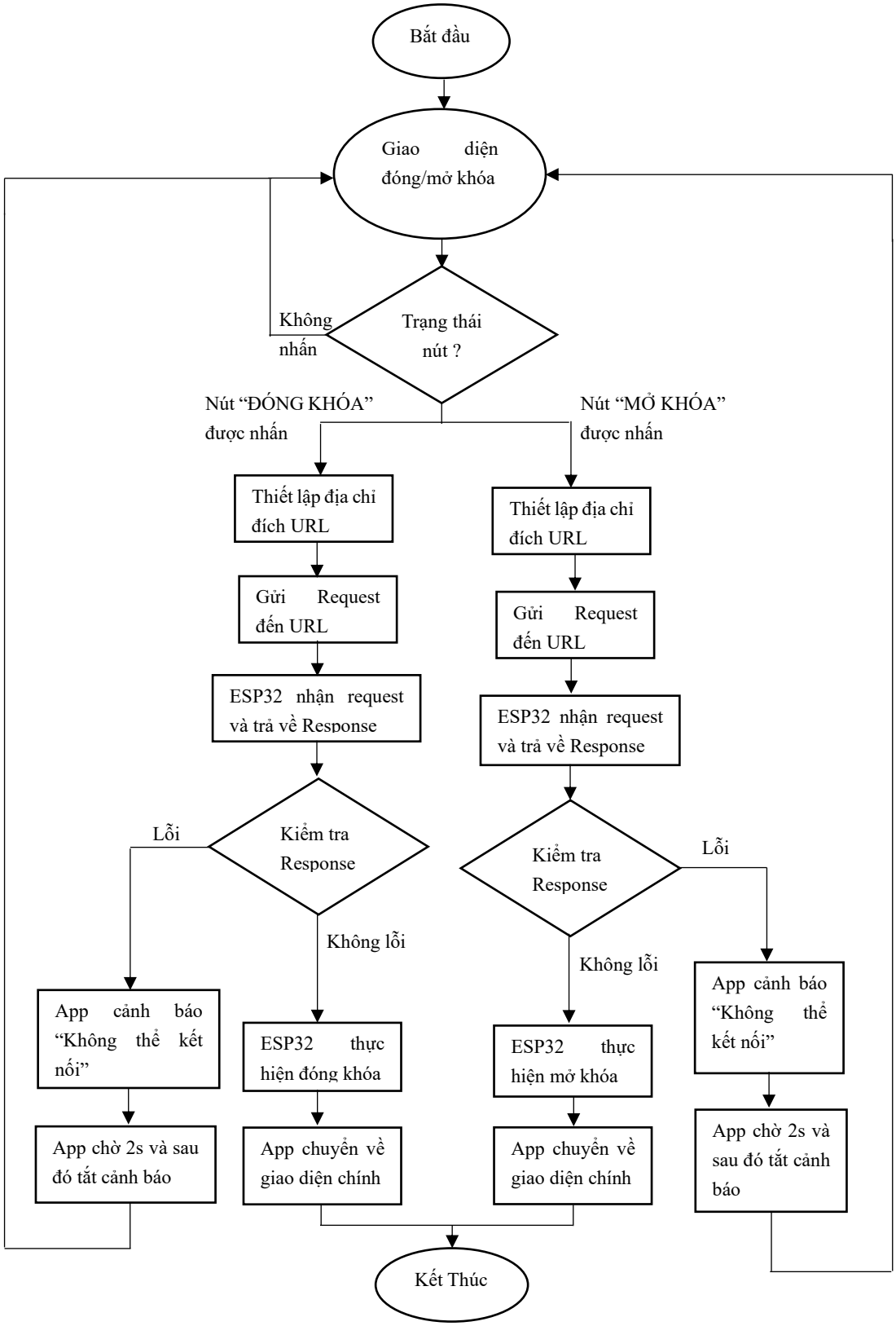
- Khối Đóng ngắt & Điều khiển Khóa (Relay, solenoid lock)

Đây là tầng thực thi cuối cùng của hệ thống, bao gồm Module Relay và Khóa điện từ (Solenoid). Khối này có nhiệm vụ chuyển đổi tín hiệu điện áp thấp (3.3V) từ vi điều khiển thành hành động cơ học thực tế. Module Relay đóng vai trò là công tắc điện tử và bộ phận cách ly, bảo vệ vi điều khiển khỏi dòng tải lớn của khóa điện. Khi nhận tín hiệu kích hoạt, Relay đóng mạch cấp nguồn cho Solenoid, thực hiện thao tác thu chốt khóa để mở cửa, hoàn tất chu trình kiểm soát ra vào.

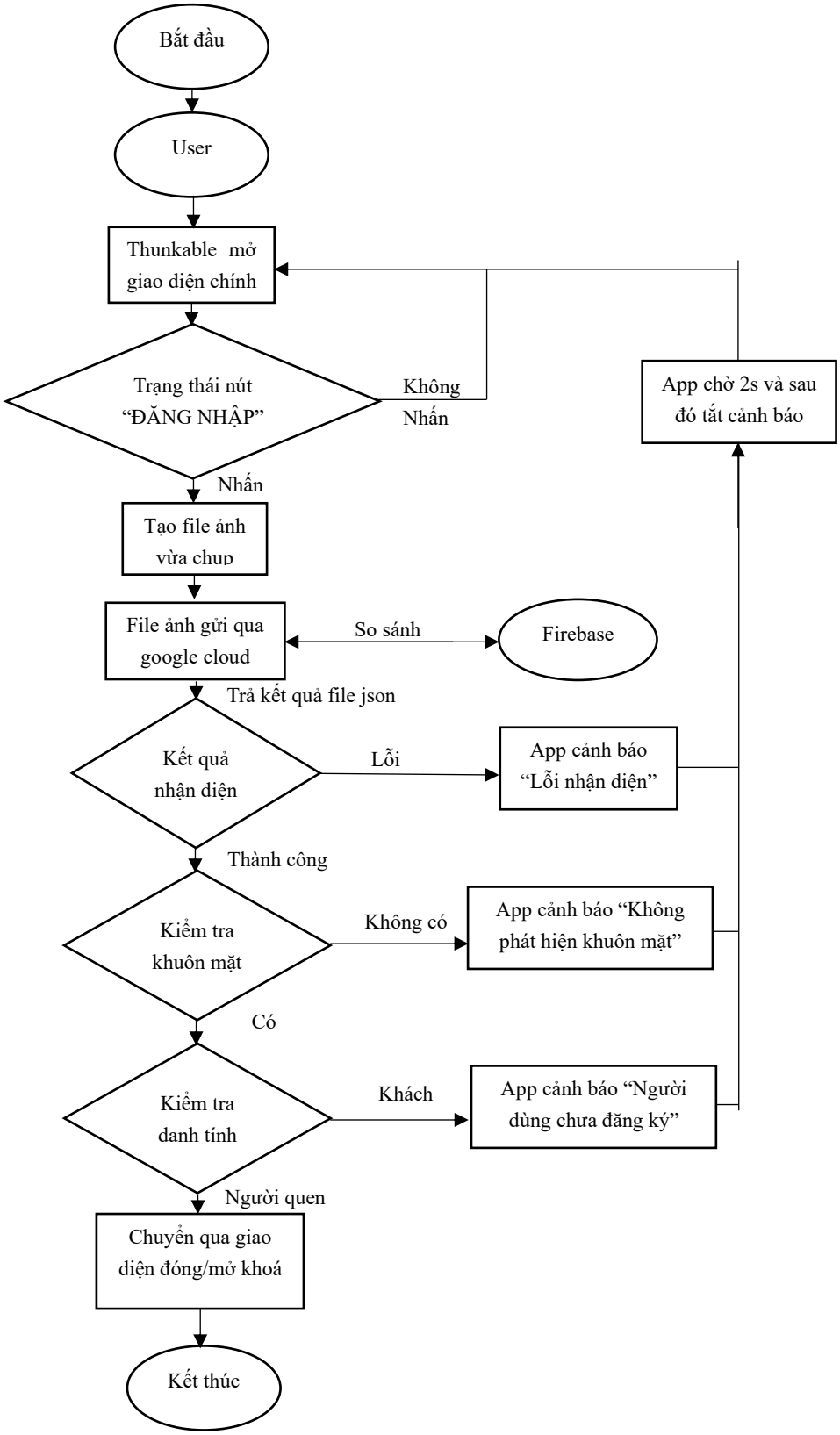
- Khối Power Supply

Đây là khối có chức năng cấp nguồn cho khối solenoid lock hoạt động. Khối cấp nguồn 12V từ adapter 12V 2A và bảo đảm cho solenoid lock hoạt động ổn định

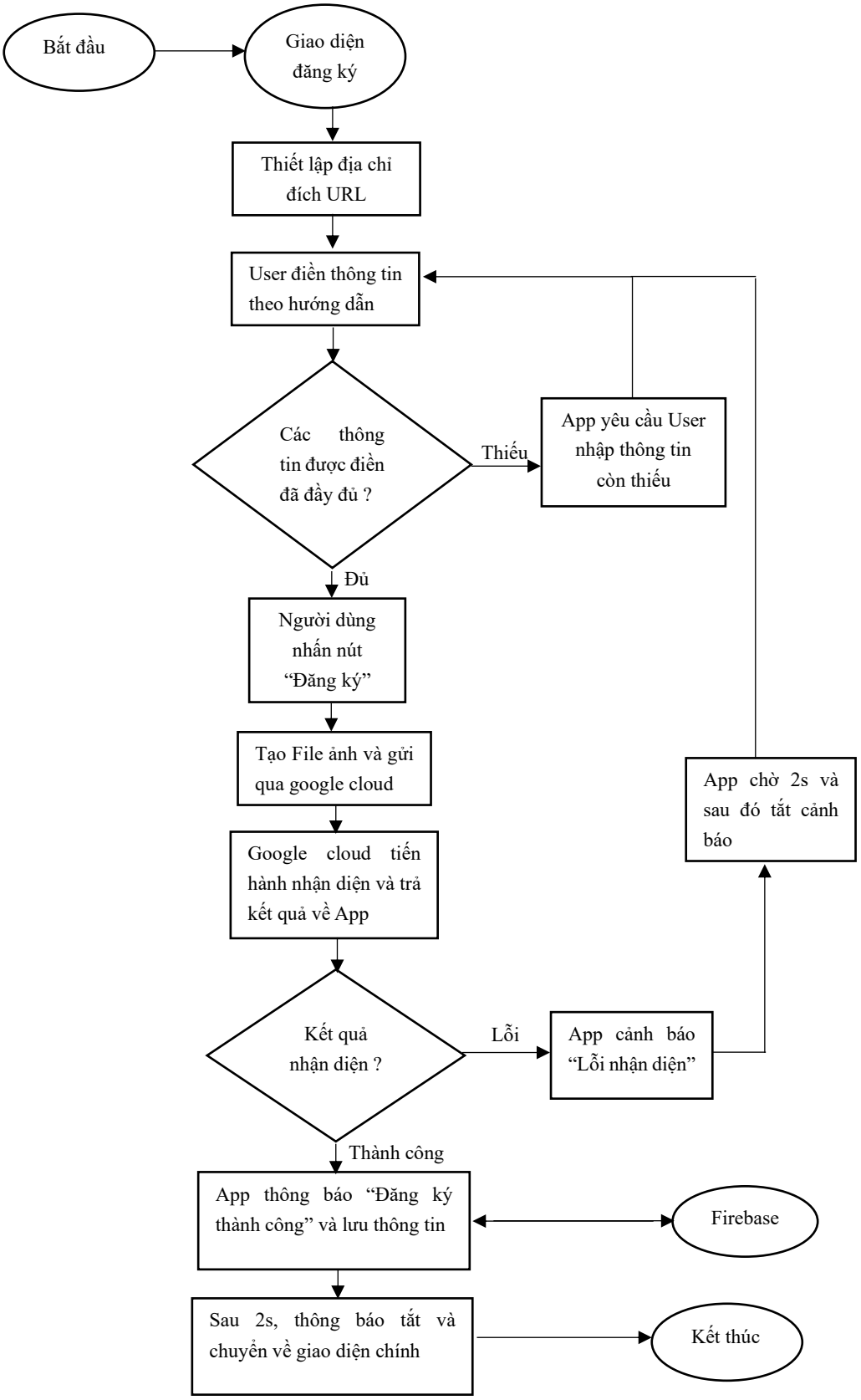
4. Quy trình đóng/mở khóa



5. Quy trình đăng nhập



6. Quy trình đăng ký



II. TỔNG QUAN LÝ THUYẾT

1. Tìm hiểu về ngôn ngữ lập trình python với AI

1.1 Giới thiệu Python

Python là một ngôn ngữ lập trình thông dịch, dễ đọc và dễ hiểu. Nền tảng nổi tiếng với cú pháp đơn giản và được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau. Điển hình như phát triển website, phân tích dữ liệu, trí tuệ nhân tạo và nhiều ứng dụng khác.

Những lợi ích Python mang đến:

Cú pháp đơn giản và dễ đọc. Cú pháp của Python rất giống với ngôn ngữ tiếng Anh tự nhiên, dễ học và dễ đọc, giúp lập trình viên tập trung vào giải quyết vấn đề hơn là việc ghi nhớ cú pháp phức tạp.

Python được sử dụng trong nhiều lĩnh vực như phát triển website, phân tích dữ liệu, trí tuệ nhân tạo, khoa học dữ liệu và nhiều ứng dụng khác. Điều này làm cho Python trở thành một ngôn ngữ lập tình linh hoạt và tiện lợi.

Cộng đồng Python khá lớn và luôn hoạt động tích cực nhằm cung cấp nhiều thư viện, framework hữu ích. Người dùng sẽ được nhận sự hỗ trợ từ cộng đồng thông qua tài liệu, diễn đàn và các nguồn thông tin trực tuyến.

Python có thể chạy trên nhiều hệ điều hành khác nhau như Windows, macOS, Linux, Raspberry Pi,… giúp cho việc phát triển và triển khai ứng dụng trở nên dễ dàng.

Python có thể dễ dàng tích hợp với các ngôn ngữ lập trình khác và các công nghệ hiện có, giúp nó trở thành một lựa chọn lý tưởng cho nhiều dự án khác nhau.

Ứng dụng của python:

Phát triển Web: Python đã củng cố vị thế là một ngôn ngữ lập trình được ưa chuộng trong lĩnh vực phát triển web, chủ yếu nhờ vào các framework hiệu suất cao như Django và Flask. Django cung cấp một bộ công cụ tích hợp toàn diện, cho phép các nhà phát triển nhanh chóng xây dựng các ứng dụng web phức tạp và giàu tính năng. Ngược lại, Flask là một micro-framework, nổi bật nhờ

tính gọn nhẹ và linh hoạt, khiến nó trở thành lựa chọn lý tưởng cho các dự án quy mô nhỏ hơn hoặc các ứng dụng đòi hỏi kiến trúc tùy biến cao.

Trí tuệ Nhân tạo và Học máy: Python đã khẳng định vị thế là ngôn ngữ chủ đạo trong lĩnh vực Trí tuệ Nhân tạo (AI) và Học máy (ML), nhờ vào hệ sinh thái thư viện vô cùng phong phú. Các nền tảng như TensorFlow và Keras cung cấp bộ công cụ toàn diện để xây dựng và triển khai các mô hình học sâu (deep learning) quy mô lớn. PyTorch lại thu hút cộng đồng nghiên cứu nhờ tính linh hoạt cao và quy trình làm việc trực quan (dynamic computation graph). Bên cạnh đó, Scikit-learn là thư viện cốt lõi cho các thuật toán học máy truyền thống, cung cấp các công cụ hiệu quả và dễ sử dụng. Lợi thế của Python nằm ở cú pháp rõ ràng, cho phép các nhà khoa học dữ liệu và kỹ sư nhanh chóng hiện thực hóa các thuật toán phức tạp với số lượng mã nguồn tối thiểu.

Internet of Things (IoT): Python là ngôn ngữ được ưa chuộng trong lĩnh vực Internet vạn vật (IoT), đặc biệt là khi triển khai trên các nền tảng như Raspberry Pi. Ngôn ngữ này đơn giản hóa đáng kể quá trình lập trình để điều khiển cảm biến, tương tác với các thiết bị phần cứng và thu thập dữ liệu từ môi trường. Nhờ cú pháp trực quan và hệ sinh thái thư viện phong phú, Python cho phép các nhà phát triển nhanh chóng hiện thực hóa các dự án IoT, từ những ý tưởng sáng tạo ban đầu đến các giải pháp phức tạp.

Xử lý Hình ảnh và Video: OpenCV (Open Source Computer Vision Library) là thư viện nền tảng và phổ biến hàng đầu trong lĩnh vực thị giác máy tính (computer vision) khi sử dụng Python. Thư viện này cung cấp một bộ công cụ toàn diện, cho phép thực thi hiệu quả các tác vụ phức tạp như nhận diện khuôn mặt, theo dõi đối tượng (object tracking), và phân tích video thời gian thực. Sự kết hợp giữa cú pháp trực quan của Python và hiệu năng mạnh mẽ của OpenCV giúp các nhà phát triển nhanh chóng xây dựng các ứng dụng xử lý hình ảnh và thị giác tinh vi.

Python là một ngôn ngữ lập trình mạnh mẽ, đa năng và dễ sử dụng, phù hợp cho cả người mới bắt đầu và các lập trình viên giàu kinh nghiệm.

1.2 Python và trí tuệ nhân tạo (AI)

Trí tuệ nhân tạo (AI) là lĩnh vực khoa học máy tính tập trung vào việc phát triển các hệ thống thông minh, có khả năng mô phỏng các tác vụ nhận thức của con người như nhận diện giọng nói, phân tích hình ảnh và ra quyết định.

Python nổi lên như một ngôn ngữ lập trình hàng đầu trong lĩnh vực AI nhờ vào những ưu điểm vượt trội:

Hệ sinh thái thư viện đa dạng: Python tự hào sở hữu một kho thư viện đồ sộ được thiết kế riêng cho AI, Machine Learning và Deep Learning, bao gồm NumPy, SciPy và scikit. Các thư viện này đóng vai trò như "lưu trữ" các hàm và đoạn mã được viết sẵn, cho phép các lập trình viên rút ngắn thời gian và công sức phát triển đáng kể. Ví dụ, NumPy được sử dụng rộng rãi cho các tính toán khoa học, SciPy xử lý các phép tính nâng cao hơn, trong khi scikit được dùng để khai thác và phân tích dữ liệu.

Cú pháp đơn giản: Python sử dụng cú pháp đơn giản, dễ đọc, giúp các nhà phát triển AI dễ dàng kiểm tra thuật toán mà không cần triển khai. Cú pháp đơn giản này cũng là chìa khóa giúp các dự án AI phát triển nhanh chóng hơn so với nhiều ngôn ngữ khác, đồng thời tạo điều kiện thuận lợi cho việc cộng tác trong các dự án phức tạp.

Cộng đồng hỗ trợ rộng lớn: Là một ngôn ngữ mã nguồn mở, Python nhận được sự hỗ trợ to lớn từ cộng đồng lập trình viên trên toàn thế giới. Các tài nguyên và tài liệu chất lượng cao luôn sẵn có, hỗ trợ đắc lực cho việc giải quyết các vấn đề phát sinh trong quá trình phát triển AI.

Khả năng tích hợp linh hoạt: Python có khả năng tích hợp mượt mà với các hệ thống được viết bằng nhiều ngôn ngữ lập trình khác, tạo điều kiện thuận lợi cho việc kết hợp với các dự án AI hiện có.

Kiểm tra nhanh chóng: Python cung cấp nhiều công cụ mạnh mẽ để kiểm tra và đánh giá mã, cho phép các nhà phát triển nhanh chóng phát hiện và sửa lỗi, đảm bảo chất lượng cho dự án AI.

Hiệu suất: Mặc dù Python bị cho là chậm hơn một số ngôn ngữ lập trình khác, giải pháp Cython đã được phát triển để khắc phục nhược điểm này. Cython

cho phép viết mã mở rộng C trực tiếp trong Python, giúp tăng tốc độ thực thi đáng kể.

Công cụ trực quan: Python cung cấp nhiều thư viện trực quan hóa dữ liệu, trong đó nổi bật là Matplotlib. Nhờ Matplotlib, các nhà khoa học dữ liệu có thể tạo biểu đồ và trình bày dữ liệu phức tạp dưới dạng hình ảnh trực quan, dễ hiểu. Tóm lại, Python với hệ sinh thái mạnh mẽ, cú pháp đơn giản, cộng đồng hỗ trợ đông đảo và khả năng tích hợp linh hoạt đã khẳng định vị thế là ngôn ngữ lập trình hàng đầu cho sự phát triển của AI.

Tóm lại, Python đã vượt qua vai trò của một ngôn ngữ lập trình đơn thuần để trở thành nền tảng cốt lõi thúc đẩy cuộc cách mạng Trí tuệ Nhân tạo. Nhờ cú pháp đơn giản, khả năng phát triển nhanh chóng và một hệ sinh thái thư viện vô song—bao gồm TensorFlow, PyTorch, và Scikit-learn—Python đã dân chủ hóa việc xây dựng các mô hình phức tạp. Sự cộng hưởng mạnh mẽ này sẽ tiếp tục là động lực chính cho những đột phá tương lai trong học sâu (deep learning), xử lý ngôn ngữ tự nhiên (NLP) và thị giác máy tính, hứa hẹn tác động sâu sắc và toàn diện đến mọi lĩnh vực, từ y tế, giáo dục đến tài chính.

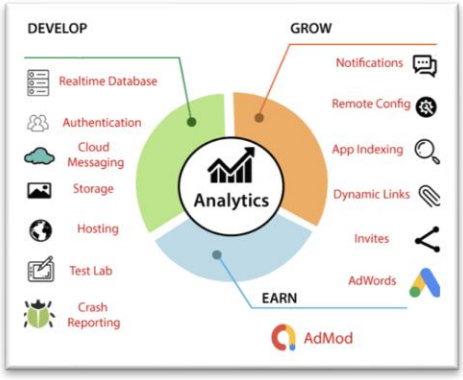
2. Tìm hiểu về Firebase

2.1 Giới thiệu Firebase

Firebase là một nền tảng phát triển ứng dụng được tạo ra bởi Google cung cấp nhiều công cụ để tạo, cải thiện và quản lý các ứng dụng web và di động. Mục tiêu chính của nó là đơn giản hóa công việc của các nhà phát triển bằng cách cung cấp các dịch vụ sẵn sàng sử dụng trong các lĩnh vực chính như cơ sở dữ liệu, xác thực, lưu trữ và phân tích.

Những tính năng của Firebase:

Firebase cung cấp một loạt các tính năng để hỗ trợ các nhà phát triển trong việc xây dựng, phát hành, giám sát, và tương tác với ứng dụng của họ. Các tính năng này được chia thành ba nhóm chính: Nhóm Chức Năng Phát Triển, Nhóm Chức Năng Theo Dõi & Phát Hành, Nhóm Chức Năng Tương Tác & Tối Ưu Hóa



Hình 6 Các tính năng của Firebase

- Nhóm Chức Năng Phát Triển

Nhóm chức năng Xây dựng cung cấp các dịch vụ nền tảng để phát triển ứng dụng một cách nhanh chóng. Các nhà phát triển có thể sử dụng Kho Dữ Liệu Đám Mây và Cơ Sở Dữ Liệu Thời Gian Thực để quản lý dữ liệu trực tuyến. Bên cạnh đó, Chức Năng Đám Mây cung cấp giải pháp máy chủ ảo, cho phép chạy mã backend để phản hồi các sự kiện hoặc yêu cầu mạng mà không cần quản lý máy chủ.

Firebase hỗ trợ các yêu cầu thiết yếu về bảo mật và quản lý tài nguyên. Xác Thực Người Dùng Firebase giúp thực hiện quy trình đăng nhập người dùng an toàn thông qua nhiều nhà cung cấp phổ biến. Lưu Trữ Đám Mây đảm bảo việc lưu trữ và quản lý an toàn các tài nguyên và nội dung do người dùng tạo ra. Ngoài ra, Học Máy Firebase cho phép tích hợp các khả năng trí tuệ nhân tạo vào ứng dụng di động.

Để hoàn thiện quy trình phát triển, Firebase cung cấp giải pháp Lưu Trữ Web có thể mở rộng cho các ứng dụng web và dịch vụ nhỏ. Đặc biệt, Bộ Công Cụ Mô Phỏng Cục Bộ cung cấp một môi trường máy tính nội bộ mạnh mẽ, cho phép tích hợp và thử nghiệm toàn bộ các tính năng Firebase khác nhau một cách hiệu quả và miễn phí trước khi triển khai chính thức.

- Nhóm Chức Năng Theo Dõi & Phát Hành

Nhóm này tập trung vào việc duy trì chất lượng ứng dụng sau khi phát hành. Hệ thống theo dõi sự cố ứng dụng là công cụ thiết yếu, cung cấp báo cáo lỗi và sự cố ngay lập tức, giúp nhanh chóng xác định và khắc phục các vấn đề

ổn định. Đồng thời, Công cụ đo lường tốc độ hoạt động theo dõi chi tiết các chỉ số hiệu năng của ứng dụng trên các nền tảng, cho phép tối ưu hóa tốc độ và trải nghiệm người dùng.

Để hỗ trợ ra quyết định dựa trên dữ liệu, Hệ thống phân tích hành vi người dùng được tích hợp sâu với Firebase, cung cấp thông tin chuyên sâu về hành vi người dùng, nguồn gốc và hiệu suất tổng thể của ứng dụng. Sự tích hợp này giúp nhà phát triển hiểu rõ cách người dùng tương tác với ứng dụng, từ đó định hướng các chiến lược phát triển và tiếp thị hiệu quả.

Firebase tối ưu hóa quá trình kiểm thử và phát hành. Cơ sở hạ tầng thử nghiệm trên Đám mây cung cấp môi trường để ứng dụng được kiểm thử tự động trên nhiều loại thiết bị và cấu hình thực tế. Sau đó, Công cụ quản lý phân phối phiên bản thử nghiệm đơn giản hóa quy trình kiểm thử phiên bản beta bằng cách quản lý và gửi các bản ứng dụng trước khi phát hành đến người thử nghiệm một cách an toàn.

- Nhóm Chức Năng Tương Tác & Tối Ưu Hóa

Nhóm chức năng này cho phép nhà phát triển linh hoạt điều chỉnh và tối ưu hóa trải nghiệm người dùng sau khi ứng dụng đã được cài đặt. Hệ thống cấu hình từ xa cho phép thay đổi giao diện và hành vi của ứng dụng ngay lập tức mà không cần yêu cầu người dùng cập nhật phiên bản mới. Đồng thời, Công cụ thử nghiệm đa biến thể A/B hỗ trợ tối ưu hóa bằng cách thử nghiệm các phiên bản khác nhau của tính năng hoặc giao diện trên các nhóm người dùng để đưa ra quyết định dựa trên dữ liệu.

Để tăng cường sự gắn kết và cá nhân hóa trải nghiệm, Firebase cung cấp các công cụ giao tiếp mục tiêu. Hệ thống gửi tin nhắn trong ứng dụng cho phép gửi các thông báo có ngữ cảnh và mục tiêu cụ thể đến người dùng đang hoạt động. Ngoài ra, Dịch vụ gửi thông báo đa nền tảng giúp gửi tin nhắn và thông báo dữ liệu miễn phí đến người dùng, ngay cả khi họ không sử dụng ứng dụng.

Firebase tận dụng trí tuệ nhân tạo để hỗ trợ các chiến lược giữ chân người dùng. Hệ thống dự đoán hành vi sử dụng thuật toán học máy để tạo các phân đoạn người dùng động dựa trên hành vi dự đoán. Cuối cùng, Đường liên kết thông minh cung cấp các liên kết giúp điều hướng người dùng đến đúng nội

dung cụ thể trong ứng dụng, góp phần cải thiện tỷ lệ chuyển đổi và tăng cường sự tham gia.

2.2 Cách hoạt động của Firebase

Firebase hoạt động như một nền tảng dịch vụ backend toàn diện, giúp nhà phát triển khỏi việc phải trực tiếp xây dựng, quản lý và vận hành cơ sở hạ tầng máy chủ phức tạp. Thay vào đó, nền tảng này cung cấp một bộ Giao diện Lập trình Ứng dụng (API) và Bộ công cụ Phát triển Phần mềm (SDK) được thiết kế để dễ dàng tích hợp, cho phép các nhà phát triển nhanh chóng bổ sung các tính năng backend vào ứng dụng của mình.

Thông qua các dịch vụ thiết yếu như cơ sở dữ liệu thời gian thực, xác thực người dùng, lưu trữ đám mây và phân tích dữ liệu, Firebase đảm nhận xử lý hầu hết các tác vụ backend quan trọng. Các dịch vụ này được xây dựng trên hạ tầng đám mây mạnh mẽ của Google, đảm bảo độ tin cậy cao, khả năng mở rộng quy mô linh hoạt theo nhu cầu người dùng và giúp tối ưu hóa chi phí vận hành.

Khi một ứng dụng kết nối với Firebase, nền tảng sẽ tự động đồng bộ hóa dữ liệu người dùng và trạng thái hệ thống ngay lập tức, đồng thời bảo vệ dữ liệu thông qua các lớp bảo mật tích hợp. Ngoài ra, Firebase còn mở rộng khả năng hoạt động với các tính năng nâng cao như gửi thông báo đẩy, thực thi các hàm máy chủ ảo và phân tích dữ liệu theo thời gian thực, cung cấp cho nhà phát triển một trải nghiệm quản lý, theo dõi và điều chỉnh ứng dụng trực tiếp, hiệu quả.

3. Tìm hiểu về Thunkable

Thunkable là một nền tảng phát triển ứng dụng di động "no-code" (không cần mã) hàng đầu, cho phép người dùng tạo ra các ứng dụng iOS và Android gốc mà không đòi hỏi kiến thức lập trình chuyên sâu. Nền tảng này hướng tới mục tiêu dân chủ hóa quá trình phát triển ứng dụng, giúp hàng triệu người dùng, từ học sinh đến doanh nghiệp, biến ý tưởng thành sản phẩm. Thunkable giúp tiết kiệm đáng kể thời gian, chi phí và tăng khả năng tiếp cận công nghệ cho tất cả mọi người.

Nền tảng này sử dụng một giao diện kéo-thả trực quan, kết hợp với hệ thống lập trình logic bằng các khối lệnh, giúp người dùng dễ dàng thiết kế và

xây dựng chức năng ứng dụng. Thunkable liên tục phát triển và cải thiện trải nghiệm người dùng. Đây là một giải pháp mạnh mẽ và đơn giản, đang thay đổi cách chúng ta tạo ra các ứng dụng di động mà không cần viết một dòng mã nào.

Chức năng nổi bật của Thunkable:

Thunkable đơn giản hóa việc phát triển ứng dụng thông qua giao diện kéo và thả (drag-and-drop) trực quan, cho phép người dùng thiết kế giao diện bằng cách sắp xếp các thành phần như nút bấm, hình ảnh và bản đồ. Thay vì viết mã, nền tảng sử dụng hệ thống lập trình bằng khối lệnh (blocks), giúp người dùng, kể cả người mới, dễ dàng ghép nối các khối logic để xử lý sự kiện và xây dựng chức năng ứng dụng.

Về khả năng kết nối, Thunkable hỗ trợ mạnh mẽ việc tích hợp API và các dịch vụ bên ngoài, cho phép ứng dụng truy cập dữ liệu từ Google Maps, Firebase hay các API của mạng xã hội. Nền tảng này cũng cung cấp các giải pháp cơ sở dữ liệu linh hoạt, từ lưu trữ nội bộ (local) đến kết nối trực tiếp với các dịch vụ đám mây như Firebase, hoặc truy cập dữ liệu tùy chỉnh thông qua API.

Nền tảng này đảm bảo trải nghiệm người dùng tối ưu thông qua thiết kế đáp ứng (responsive design), giúp giao diện tự động điều chỉnh phù hợp với nhiều kích thước màn hình, từ điện thoại đến máy tính bảng. Cuối cùng, Thunkable hỗ trợ quy trình kiểm thử và phát hành toàn diện, cho phép người dùng kiểm tra ứng dụng trực tiếp trên thiết bị thật và dễ dàng xuất bản sản phẩm lên cả Google Play Store và App Store.

4. Nhận diện khuôn mặt

Nhận diện khuôn mặt là một phương pháp sử dụng công nghệ để xác định hoặc xác nhận danh tính của một cá nhân qua khuôn mặt của họ. Hệ thống nhận diện khuôn mặt có thể được sử dụng để nhận diện người trong các bức ảnh, video hoặc trong thời gian thực.

Bài toán này được chia ra thành 2 vấn đề chính là Phát hiện khuôn măt (Face Detection) và Phân biệt khuôn mặt (Face Verification). Để giải quyết hiệu quả chuỗi bài toán phức tạp này, các hệ thống hiện đại thường triển khai một

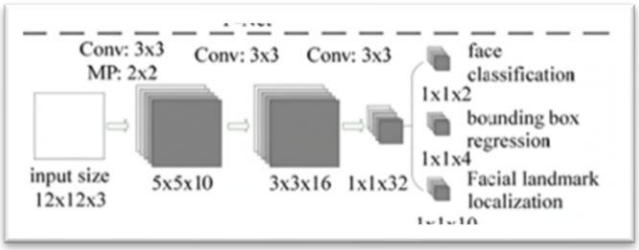
quy trình kết hợp hai kiến trúc mạng nơ-ron tích chập là MTCNN (Multi-task Cascaded Convolutional Networks) và FaceNet.

4.1 Mô hình MTCNN (Multi-task Cascaded Convolutional Networks)

Phần đầu tiên sẽ là về Face Detection, một bài toán với nhiệm vụ phát hiện các khuôn mặt có trong ảnh hoặc frame trong Video. MTCNN đóng vai trò then chốt trong giai đoạn tiền xử lý, chịu trách nhiệm phát hiện (detection) và căn chỉnh (alignment) khuôn mặt.

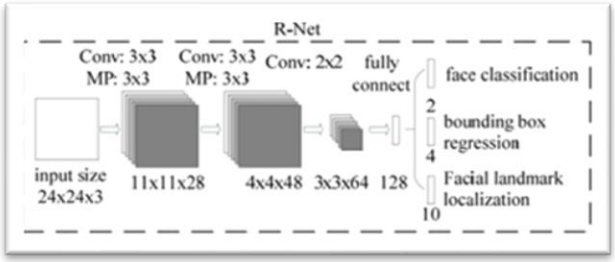
Kiến trúc của mô hình:

Về mặt kiến trúc, MTCNN bao gồm ba mạng nơ-ron được kết nối theo kiểu xếp tầng (cascade), bao gồm: P-Net, R-Net và O-Net. Để đảm bảo hệ thống có thể nhận diện các khuôn mặt ở nhiều kích thước khác nhau, ảnh đầu vào gốc sẽ được tái tạo (rescaled) thành nhiều phiên bản với các độ phân giải khác nhau trước khi đưa vào xử lý.



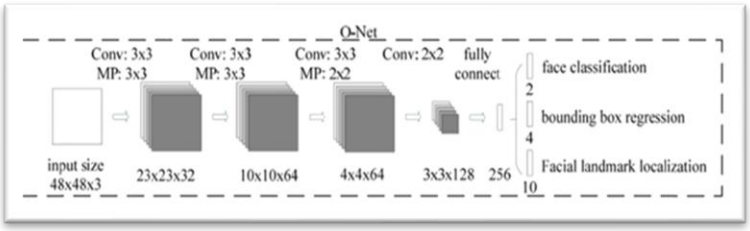
Hình 7 Kiến trúc mạng P-Net

Kết quả từ P-Net vẫn còn tương đối thô và chứa nhiều dự đoán sai (false positives). Do đó, các kết quả này được đưa vào R-Net để tinh chỉnh thêm. R-Net sẽ loại bỏ phần lớn các cửa sổ sai và tiếp tục sử dụng hồi quy hộp giới hạn cùng NMS để cải thiện độ chính xác của các dự đoán



Hình 8 Kiến trúc mạng R-Net

Cuối cùng, các ứng viên còn lại sau khi qua R-Net sẽ được đưa vào O-Net. Mạng này thực hiện bước tinh chỉnh cuối cùng để xuất ra vị trí khung khuôn mặt (face frame) chính xác nhất. Điểm khác biệt quan trọng của O-Net là ngoài việc dự đoán hộp giới hạn, nó còn dự đoán vị trí của 5 điểm mốc đặc trưng (facial landmarks) của khuôn mặt (thường là hai mắt, chóp mũi và hai khóe miệng).



Hình 9 Kiến trúc mạng O-Net

Mỗi mạng trong MTCNN có ba phần đầu ra (output), vì vậy hàm mất mát (loss) cũng được cấu thành từ ba thành phần. Đối với phần phát hiện khuôn mặt, ta sử dụng trực tiếp hàm mất mát cross entropy:

$$L_i^{det} = -(y_i^{det} \log(p_i) + \left(1 - y_i^{det}\right)(1 - \log(p_i)))$$

Trong công thức, p_i đại diện cho xác suất đầu vào là một khuôn mặt, và y_i^{det} là nhãn thực (true label).

Đối với bài toán hồi quy hộp (box regression) và bài toán quyết định năm điểm đặc trưng (feature point decisions), cả hai đều là các bài toán hồi quy (regression), vì vậy ta sử dụng khoảng cách Euclidean (Euclidean distance) thông dụng để tính toán mất mát. Hàm mất mát cho hồi quy hộp giới hạn:

$$L_i^{box} = \left\| \hat{y}_i^{landmark} - y_i^{landmark} \right\|_2^2$$

Trong đó $\hat{y}_i^{landmark}$ là (tọa độ) được dự đoán bởi mạng, và $y_i^{landmark}$ là tọa độ thực tế (ground truth) của hộp. Cuối cùng, ba hàm mất mát này sẽ được nhân với trọng số (weights) riêng của chúng và sau đó được cộng lại với nhau để tạo thành hàm mất mát tổng (total loss) cuối cùng.

Tổng quan về hiệu năng và tính ứng dụng của MTCNN:

Thế mạnh đầu tiên của MTCNN là kiến trúc đa tác vụ (multi-tasking), cho phép mô hình thực hiện đồng thời nhiều nhiệm vụ. Cụ thể, MTCNN không

chỉ phát hiện hộp giới hạn (bounding box) mà còn định vị chính xác các điểm mốc (landmarks) như mắt, mũi và miệng. Việc trích xuất đồng thời các thông tin này tạo tiền đề vững chắc cho các ứng dụng phức tạp hơn như nhận dạng, phân loại thuộc tính hay phân tích biểu cảm.

Độ tin cậy cao là một ưu điểm cốt lõi, xuất phát từ việc sử dụng nền tảng mạng nơ-ron tích chập (CNN) mạnh mẽ. Kiến trúc này đã được chứng minh hiệu quả trong các tác vụ xử lý ảnh, giúp MTCNN đạt được độ chính xác ấn tượng. Đáng chú ý, mô hình thể hiện khả năng kháng vượt trội trước các thách thức thực tế như nhiễu ảnh, thay đổi điều kiện ánh sáng và sự đa dạng về tỷ lệ khuôn mặt.

Về hiệu suất, MTCNN được thiết kế để hoạt động nhanh và tối ưu. Kiến trúc xếp tầng (cascade) cho phép mô hình xử lý tuần tự qua các mạng P-Net, R-Net và O-Net, giúp loại bỏ nhanh các dự đoán sai và tập trung tài nguyên tính toán. Nhờ vậy, MTCNN có khả năng xử lý hình ảnh và video với độ phức tạp đa dạng, đáp ứng tốt các yêu cầu về tốc độ trong nhiều ứng dụng thực tiễn.

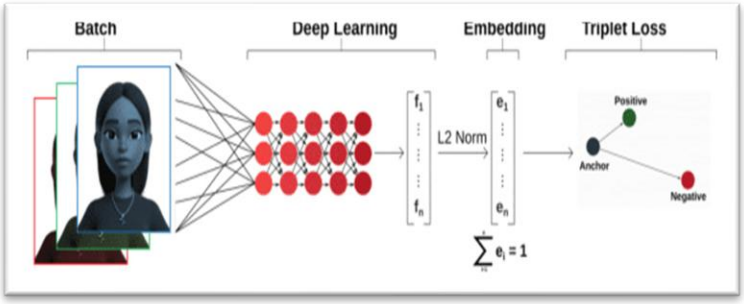
Cuối cùng, với khả năng ứng dụng đa dạng, MTCNN là thành phần nền tảng cho nhiều bài toán từ nhận dạng danh tính, phân loại thuộc tính đến giám sát an ninh. Cấu trúc ba tác vụ (phát hiện, định vị điểm mốc và hồi quy) giúp mô hình xử lý hiệu quả các ảnh phức tạp, có sự thay đổi về kích thước, tư thế và ánh sáng, mang lại kết quả hứa hẹn trên cả ảnh tĩnh lẫn video. Trong khuôn khổ nghiên cứu này, MTCNN được triển khai cho giai đoạn đầu: phát hiện và trích xuất (crop) vùng khuôn mặt, làm dữ liệu đầu vào chuẩn hóa cho tác vụ nhận dạng người dùng.

4.2 Mô hình FaceNet

FaceNet là mô hình học sâu do Google giới thiệu năm 2015, sử dụng mạng CNN để giải quyết bài toán nhận dạng bằng cách ánh xạ khuôn mặt vào một không gian đặc trưng. Thay vì phân loại, mô hình học cách tạo ra một vector 128 chiều cho mỗi ảnh. Trong không gian này, khoảng cách Euclidean giữa các vector tương ứng trực tiếp với độ tương đồng. Các khuôn mặt của cùng một người sẽ ở gần nhau.

Để đạt được khả năng phân biệt cao, FaceNet được tối ưu hóa bằng một hàm mất mát đặc thù là Triplet Loss. Phương pháp huấn luyện này đảm bảo các vector embedding có tính đại diện cao và tách biệt rõ ràng giữa các cá nhân khác nhau. Nhờ vậy, các vector này có thể được sử dụng hiệu quả cho nhiều tác vụ như xác thực, tìm kiếm và phân cụm khuôn mặt.

Kiến trúc của mô hình:



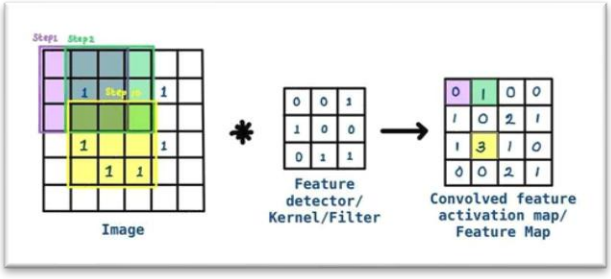
Hình 10 Kiến trúc mô hình FaceNet

Mô hình FaceNet sở hữu một kiến trúc học sâu phức tạp, được thiết kế để tối ưu hóa việc trích xuất và ánh xạ đặc trưng khuôn mặt. Cấu trúc này bao gồm các thành phần then chốt sau:

- Khối trích xuất đặc trưng (CNN)

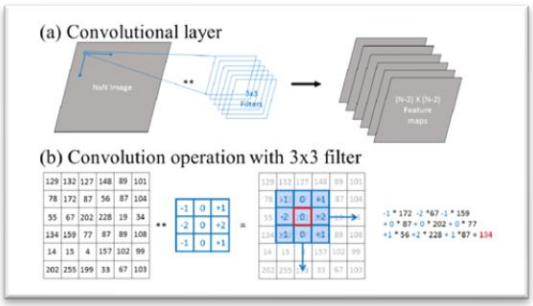
Nền tảng của FaceNet là một mạng nơ-ron tích chập (CNN) sâu, hoạt động như một khối trích xuất đặc trưng. Khối này bao gồm một chuỗi các lớp Tích chập (Convolutional Layers) để học các đặc trưng phân cấp của khuôn mặt, từ các chi tiết cơ bản như cạnh và kết cấu, đến các cấu trúc phức tạp như hình dáng mắt và mũi.

Với dữ liệu đầu vào là một hình ảnh màu, nó sẽ được biểu diễn dưới dạng một ma trận các pixel trong không gian ba chiều (3D), với ba chiều tương ứng là chiều cao, chiều rộng và độ sâu (tương ứng với RGB trong hình ảnh).



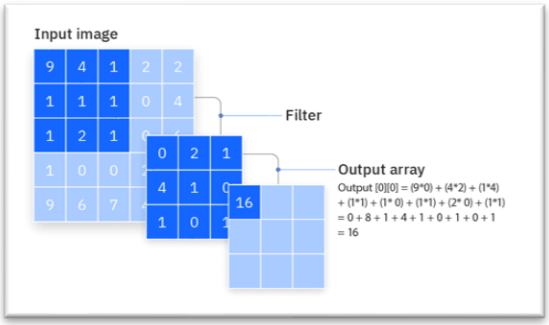
Hình 11 Phép tích chập

Feature detector (còn gọi là kernel hoặc filter) là một ma trận trọng số hai chiều (2D) giúp phát hiện các đặc trưng trong hình ảnh. Khi filter này di chuyển qua các receptive fields (vùng cảm nhận) của hình ảnh, nó sẽ kiểm tra xem các đặc trưng cần tìm có xuất hiện hay không. Quá trình này được gọi là convolution.



Hình 12 Quá trình tích chập với 3×3 filter

Feature detector thường có kích thước nhỏ, ví dụ như ma trận 3×3, và nó xác định kích thước của receptive field. Khi filter này di chuyển qua hình ảnh, một tích vô hướng (dot product) được tính giữa các giá trị pixel trong vùng quét và filter. Kết quả này sẽ được đưa vào một mảng đầu ra. Sau đó, filter dịch chuyển theo một khoảng cách xác định (gọi là stride) và quá trình được lặp lại cho đến khi filter đã quét qua toàn bộ hình ảnh.



Hình 13 Feature detector

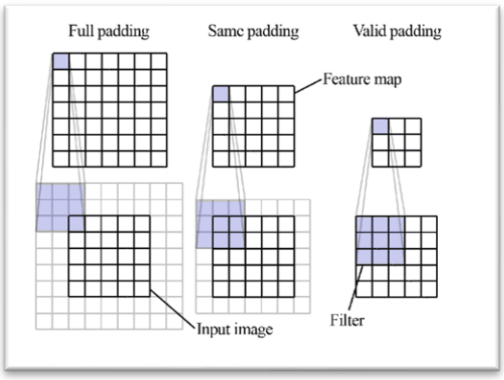
Feature map (hay activation map hoặc convolved feature) chính là kết quả của chuỗi các phép tính tích vô hướng giữa pixel đầu vào và filter.

Một điểm quan trọng là các trọng số trong feature detector không thay đổi khi nó di chuyển qua hình ảnh, đây được gọi là parameter sharing. Trong

quá trình huấn luyện, trọng số sẽ được điều chỉnh thông qua backpropagation và gradient descent.

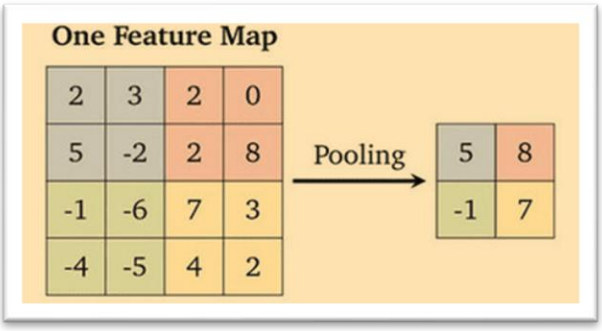
Kích thước của bản đồ đặc trưng đầu ra (feature map) được quyết định bởi ba siêu tham số chính là số lượng filters (bộ lọc) quy định độ sâu (số kênh) của đầu ra, Stride (bước di chuyển) ảnh hưởng đến kích thước không gian và cuối cùng là Zero-padding (đệm số không) được dùng để quản lý kích thước biên, với các chế độ phổ biến như thu hẹp hay giữ nguyên kích thước đầu vào.

Sau mỗi phép toán tích chập, hàm kích hoạt phi tuyến ReLU (Rectified Linear Unit) được áp dụng lên bản đồ đặc trưng. Bước này rất quan trọng vì nó đưa tính phi tuyến tính (non-linearity) vào mô hình. Điều này cho phép mạng nơ-ron học được các mối quan hệ phức tạp và các biểu diễn dữ liệu bậc cao, thay vì chỉ bị giới hạn trong các phép biến đổi tuyến tính



Hình 14 Valid padding, Same padding và Full padding

Để tăng hiệu quả, các lớp Max Pooling được xen kẽ nhằm giảm kích thước không gian và tăng tính bất biến cho đặc trưng. Khi filter di chuyển qua đầu vào, nó chọn pixel có giá trị lớn nhất trong vùng quét và đưa giá trị đó vào mảng đầu ra. Đây là phương pháp phổ biến nhất trong các mạng nơ-ron. Mặc



dù Pooling Layer làm giảm thông tin trong dữ liệu nhưng nó cũng giảm độ phức tạp tính toán, cải thiện hiệu suất và đặc biệt hạn chế hiện tượng overfitting

Hình 15 Max Pooling

- Lớp ánh xạ Embedding (Fully Connected Layer)

Sau khi đi qua khối CNN, bản đồ đặc trưng (feature map) được làm phẳng (flattened) và đưa vào một hoặc nhiều lớp kết nối đầy đủ (Fully Connected - FC). Nhiệm vụ của các lớp này là ánh xạ các đặc trưng bậc cao đã học được thành một vector biểu diễn (embedding) nén, có kích thước cố định. Trong FaceNet, vector embedding này thường có 128 chiều. Vector đầu ra này sau đó được chuẩn hóa L2 (L2 normalization) để nằm trên một mặt cầu đơn vị, giúp việc tính toán khoảng cách Euclide trở nên hiệu quả và tương đương với độ tương đồng cosine.

- Hàm mất mát Triplet Loss

Để tối ưu hóa các vector embedding, FaceNet sử dụng hàm mất mát Triplet Loss. Đây là thành phần cốt lõi giúp mô hình học được một không gian đặc trưng có tính phân biệt cao. Quá trình huấn luyện yêu cầu các bộ ba (triplets) dữ liệu, bao gồm: Anchor - x^a, ảnh đầu ra của mạng, Positive - x^p, cùng người với Anchor và Negative - x^n, khác người với Anchor.

Mục tiêu của hàm mất mát là điều chỉnh các trọng số của mạng sao cho khoảng cách Euclide giữa Anchor và Positive nhỏ hơn khoảng cách giữa Anchor và Negative, cộng thêm một khoảng lề (margin) α

Hàm mất mát L được định nghĩa như sau:

$$L = \sum_{i}^{N} [\|f(x_i^A) - f(x_i^P)\|_2^2 - \|f(x_i^A) - f(x_i^N)\|_2^2 + \alpha$$

Trong đó

f(x) là hàm ánh xạ (mạng CNN) chuyển ảnh x thành vector embedding.

α là siêu tham số lề (margin) để kiểm soát khoảng cách tối thiểu.

Bằng cách tối thiểu hóa hàm mất mát này, mô hình bị "buộc" phải đẩy các embedding của người khác nhau (Anchor và Negative) ra xa nhau, đồng thời kéo các embedding của cùng một người (Anchor và Positive) lại gần nhau, với một khoảng cách an toàn ít nhất là α.

- Kỹ thuật tăng cường dữ liệu (Data Augmentation)

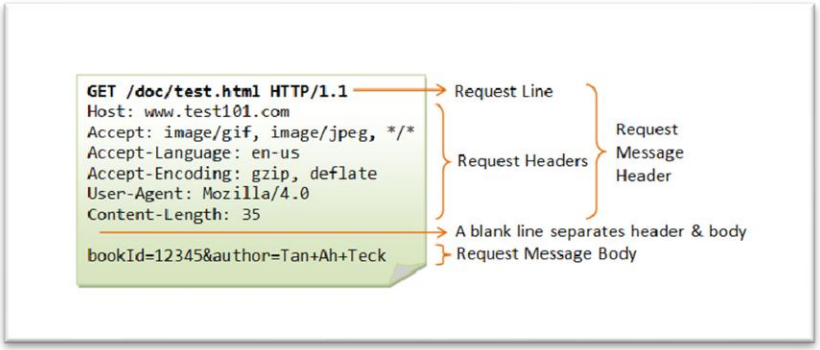
Để tăng cường khả năng tổng quát hóa và giảm thiểu hiện tượng học vẹt (overfitting), FaceNet sử dụng các kỹ thuật tăng cường dữ liệu (Data Augmentation) trong quá trình huấn luyện. Các phép biến đổi phổ biến được áp dụng bao gồm biến đổi màu sắc (độ sáng, tương phản), biến dạng hình học (xoay, cắt xén, thay đổi tỷ lệ), và thêm nhiễu (Gaussian, làm mờ). Việc tạo ra các phiên bản dữ liệu đa dạng này giúp mô hình trở nên kháng (robust) hơn với các điều kiện thực tế như thay đổi ánh sáng hay góc chụp, từ đó cải thiện đáng kể hiệu suất nhận diện cuối cùng.

5. Cơ chế giao tiếp HTTP giữa Thunkable và ESP32

HTTP request là thông tin được gửi từ client lên server, để yêu cầu server tìm hoặc xử lý một số thông tin, dữ liệu mà client muốn. HTTP request có thể là một file text dưới dạng XML hoặc Json mà cả hai đều có thể hiểu được.

5.1 Cấu trúc của HTTP Request

Bất cứ HTTP Request nào cũng có cấu trúc cụ thể. Nó gồm 3 thành phần chính. Đó là Request line, Request header và Body Request.



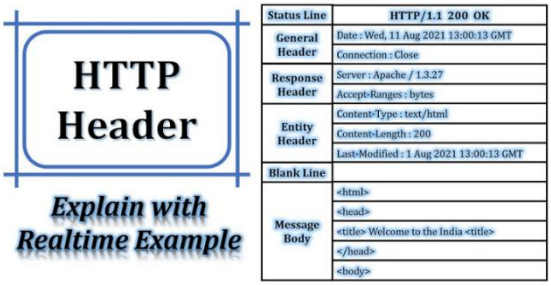
Hình 16 Các thành phần trong cấu trúc của Http Request

- Request line

Request Line là thành phần khởi đầu trong cấu trúc của một gói tin HTTP Request. Nhiệm vụ chính của dòng này là cung cấp cho máy chủ các thông tin cơ bản nhất về hành động mà client (máy khách) mong muốn thực hiện. Cấu trúc của một Request Line bao gồm ba yếu tố cốt lõi là Phương thức HTTP được sử dụng, thành phần URL giúp máy chủ xác định các tài nguyên mà máy khách yêu cầu và phiên bản của giao thức internet HTTP.

- Request header

HTTP Request Header là tập hợp các trường thông tin bổ trợ (metadata) đi kèm sau Request Line, đóng vai trò cầu nối quan trọng giúp chuyển tải ngữ cảnh chi tiết của yêu cầu từ client đến server. Các thông số trong header (Header Parameters) cho phép máy chủ hiểu rõ hơn về môi trường của client, từ đó xử lý và phản hồi dữ liệu với định dạng và cấu hình tối ưu nhất.



Hình 17 Ví dụ về Request header giúp gửi các yêu cầu từ client đến server

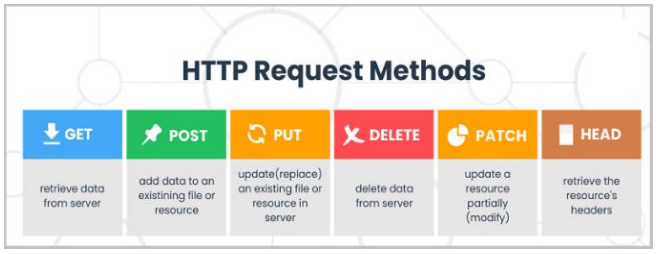
- Body Request

HTTP Request Body (hay còn gọi là Payload) là thành phần chứa dữ liệu thực tế mà client (máy khách) gửi đến server (máy chủ). Chức năng chính của Request Body là bổ sung các dữ liệu chi tiết mà các tham số trong Header hoặc URL (Header Parameters/Query Strings) không thể truyền tải hết hoặc không đảm bảo tính bảo mật. Nhờ có Request Body, client có thể gửi các thông tin như biểu mẫu đăng ký, tệp tin tải lên, hoặc các cấu trúc dữ liệu JSON/XML để server xử lý.

5.2 Các phương thức chính trong phương pháp HTTP Request

Giao thức HTTP cung cấp một tập hợp các phương thức chuẩn. Các phương thức này được khai báo ngay tại thành phần Request Line và đóng vai

trò chỉ thị trực tiếp cho server biết cần phải xử lý dữ liệu như thế nào (ví dụ: truy xuất, tạo mới, cập nhật hay xóa bỏ).



Hình 18 Các phương thức trong HTTP Request

Trong đó các phương thức thường được sử dụng:

- GET

Phương thức phổ biến nhất dùng để truy xuất dữ liệu từ máy chủ, trong đó các tham số được đính kèm trực tiếp vào URL (Query String) thay vì chứa trong phần thân yêu cầu. Do đặc thù này, GET chỉ hỗ trợ định dạng chuỗi ASCII và bị giới hạn dung lượng theo chiều dài tối đa của URL.

Tuy nhiên, ưu điểm lớn nhất của phương thức này là khả năng tận dụng bộ nhớ đệm (cache), lưu lịch sử truy cập và hỗ trợ đánh dấu trang (bookmark), giúp tối ưu tốc độ cho các tác vụ đọc dữ liệu thông thường.

- POST

Ngược lại với GET, phương thức POST được thiết kế để gửi dữ liệu nhằm tạo mới tài nguyên hoặc xử lý tác vụ, với toàn bộ thông tin được đóng gói kín trong phần thân yêu cầu (Request Body). Cơ chế này giúp dữ liệu không hiển thị trên thanh địa chỉ hay lịch sử trình duyệt, tăng cường tính bảo mật cho các thông tin nhạy cảm. Đồng thời, POST cho phép gửi dữ liệu với dung lượng lớn không giới hạn và hỗ trợ đa dạng định dạng từ văn bản đến dữ liệu nhị phân (như hình ảnh, video).

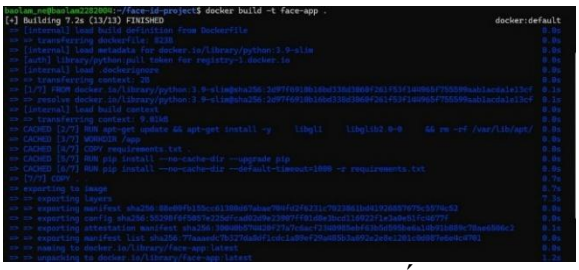
- PUT

Phương thức PUT hoạt động trên cơ chế truyền tải tương tự POST nhưng mang mục đích chuyên biệt là cập nhật tài nguyên đã tồn tại. Đặc điểm cốt lõi của PUT là cơ chế thay thế hoàn toàn (Replacement); nghĩa là máy khách cần gửi toàn bộ dữ liệu mới của đối tượng để máy chủ ghi đè lên phiên bản cũ. Trong trường hợp tài nguyên đích chưa tồn tại, tùy vào cấu hình hệ thống, PUT có thể đóng vai trò khởi tạo mới tài nguyên đó.

III. KẾT QUẢ THIẾT KẾ HỆ THỐNG NHẬN DIỆN

1. Đóng gói ứng dụng Backend (Containerization)

Quá trình khởi tạo và đóng gói môi trường thực thi cho phía Server (Backend) sử dụng Docker.



Hình 19 Quá trình đóng gói ứng dụng và thiết lập môi trường thực thi với Docker

Quá trình đóng gói ứng dụng được tối ưu hóa thông qua Docker với base image python:3.9-slim nhằm giảm thiểu tài nguyên tiêu thụ, đồng thời tích hợp cài đặt tự động các thư viện đồ họa thiết yếu (như libgl1) để đảm bảo OpenCV hoạt động ổn định trên nền tảng Linux. Việc quản lý các gói phụ thuộc qua requirements.txt kết hợp với cơ chế build tự động đã tạo ra một Docker Image hoàn chỉnh (face-app:latest), đảm bảo tính nhất quán tuyệt đối của môi trường thực thi, giúp hệ thống vận hành đồng bộ và chính xác khi triển khai từ máy phát triển (Local) lên Google Cloud Run.

2. Triển khai hệ thống nhận diện trên Google Cloud

Sau quá trình đóng gói, hệ thống được triển khai lên môi trường thực tế (Production) sử dụng dịch vụ Google Cloud.

```
baolam_ne@baolam2282004:~/face-id-project$ gcloud run deploy face-api --image gcr.io/cloud-b5047/face-app:v2
```

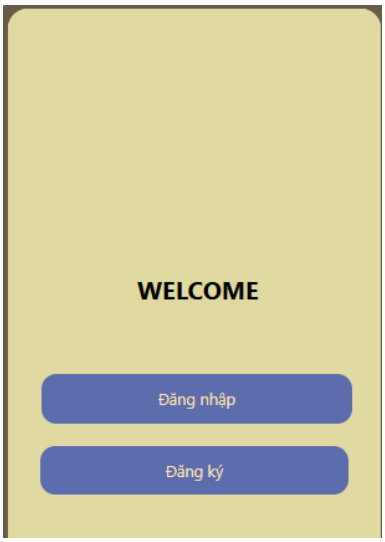
Hình 20 Quá trình Deploy Container Image từ Google Container Registry (GCR) lên môi trường Serverless

Cho phép khởi tạo dịch vụ có tên face-api. Hệ thống tự động trích xuất Docker Image phiên bản v2 được lưu trữ trên Google Container Registry (gcr.io) để thiết lập môi trường serverless. Phương pháp triển khai này giúp hệ thống tận dụng khả năng tự động mở rộng của Google Cloud, đồng thời dễ dàng quản lý phiên bản khi có cập nhật mới mà không cần can thiệp thủ công vào hạ tầng máy chủ.

3. App Mobile

Để hiện thực hóa quy trình nhận diện khuôn mặt trên nền tảng di động, giao diện ứng dụng được thiết kế theo luồng tương tác tuần tự, đảm bảo người dùng có thể thao tác dễ dàng ngay từ lần đầu sử dụng. Hệ thống giao diện bao gồm ba màn hình chính

3.1 Giao diện chính



Hình 21 Giao diện chính

Tại giao diện khởi động, hệ thống cung cấp hai tùy chọn chức năng chính là Đăng nhập và Đăng ký. Đối với quy trình Đăng nhập, khi người dùng kích hoạt chức năng, hệ thống sẽ khởi động module Camera để thực hiện thu thập mẫu khuôn mặt.

Dựa trên kết quả trả về từ phía Server, hệ thống thực hiện cơ chế rẽ nhánh. Với trường hợp hợp lệ, nếu khuôn mặt trùng khớp với dữ liệu đã đăng ký, ứng dụng tự động điều hướng sang Giao diện Điều khiển, cho

phép người dùng thực hiện thao tác Đóng/Mở khóa. Với trường hợp không hợp lệ, nếu không tìm thấy dữ liệu trùng khớp, hệ thống sẽ hiển thị thông báo cảnh báo 'Người dùng chưa đăng ký' và từ chối truy cập.

Ngoài ra, nếu người dùng lựa chọn chức năng Đăng ký, ứng dụng sẽ chuyển hướng sang giao diện đăng ký để người dùng mới thực hiện quy trình thêm dữ liệu khuôn mặt vào hệ thống.

3.2 Giao diện đăng ký

Chức năng đăng ký đóng vai trò thu thập dữ liệu mẫu để khởi tạo hồ sơ người dùng, bao gồm thông tin hành chính và dữ liệu sinh trắc học (khuôn mặt) phục vụ huấn luyện mô hình nhận diện. Module này được thiết kế nhằm đảm bảo tính chính xác và đầy đủ của dữ liệu đầu vào ngay từ bước đầu, đồng thời cân bằng hợp lý giữa yếu tố bảo mật chặt chẽ và trải nghiệm người dùng trực quan

Hình 22 Giao diện đăng ký

Giao diện đăng ký được thiết kế tối giản với bố cục mạch lạc. Phần trên hiển thị hướng dẫn quy trình giúp người dùng dễ dàng thao tác, phần trung tâm là khu vực nhập liệu gồm Tên và Mật khẩu. Việc yêu cầu mật khẩu kết hợp với nhận diện khuôn mặt nhằm thiết lập cơ chế bảo mật hai lớp, giúp tăng cường đáng kể độ an toàn cho tài khoản người dùng.

Tính năng cốt lõi là nút "Chụp ảnh", cho phép hệ thống kích hoạt camera để thu thập mẫu khuôn mặt và trích xuất đặc trưng cho thuật toán

Face-ID. Hệ thống tích hợp cơ chế kiểm tra tính toàn vẹn dữ liệu (validation) chạy ngầm, đảm bảo người dùng cung cấp đầy đủ thông tin và hình ảnh trước khi xử lý. Nếu phát hiện thiếu sót, ứng dụng sẽ gửi cảnh báo ngay lập tức để ngăn chặn việc tạo ra các bản ghi dữ liệu không hợp lệ.

Sau khi người dùng xác nhận, dữ liệu được đóng gói và gửi về hệ thống Backend/Cloud để xử lý. Ứng dụng sẽ hiển thị thông báo trạng thái ("Thành công" hoặc "Lỗi") dựa trên phản hồi từ máy chủ. Nếu đăng ký thành công, hệ thống tự động điều hướng về giao diện chính, cho phép người dùng bắt đầu phiên đăng nhập bằng tính năng nhận diện khuôn mặt (Face-ID) dựa trên dữ liệu vừa khởi tạo.

3.3 Giao diện đóng/mở khóa

Giao diện đóng/mở khóa đóng vai trò là trung tâm chỉ huy của ứng dụng, được kích hoạt ngay khi hệ thống hoàn tất quy trình xác thực sinh trắc học thành công. Tại phân hệ này, người dùng được trao quyền kiểm soát trực tiếp và tức thời đối với cơ cấu chấp hành của hệ thống (khóa cửa). Mục tiêu của giao diện là tạo ra cầu nối tương tác thời gian thực giữa người dùng và thiết bị phần cứng, đảm bảo các lệnh an ninh được thực thi chính xác.

Hình 23 Giao diện đóng/mở khóa

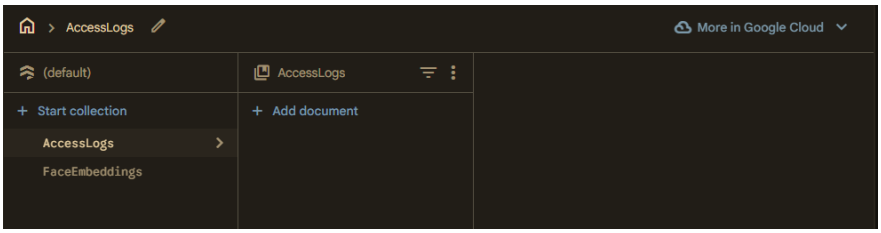
Về mặt chức năng, hệ thống cung cấp hai cơ chế điều khiển cốt lõi. Đối với chức năng "Mở khóa", ngay khi người dùng tương tác với nút lệnh, ứng dụng sẽ lập tức phản hồi bằng thông báo trạng thái "MỞ KHÓA" trên màn hình

để xác nhận hành động, đồng thời truyền tải một tín hiệu điều khiển mã hóa qua môi trường mạng đến bộ xử lý trung tâm ESP32 để kích hoạt rơ-le mở chốt. Quy trình tương tự được áp dụng cho chức năng "Đóng khóa"; khi lệnh này được kích hoạt, hệ thống sẽ gửi yêu cầu đóng chốt cửa về vi điều khiển, đảm bảo tái lập trạng thái an ninh cho khu vực giám sát một cách nhanh chóng.

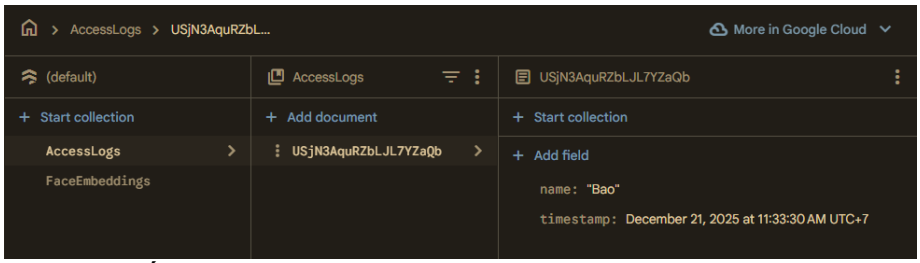
Thiết kế của giao diện này tối giản hóa các thành phần, loại bỏ các yếu tố gây nhiễu để người dùng tập trung hoàn toàn vào thao tác điều khiển thiết bị một cách nhanh chóng và chính xác nhất.

4. Cơ sở dữ liệu (database)

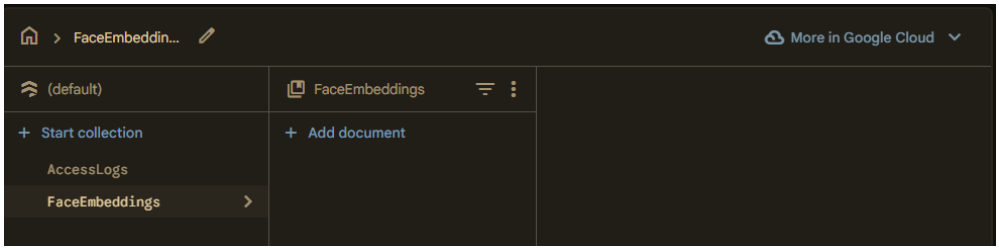
Hệ thống sử dụng nền tảng Firebase làm cơ sở dữ liệu trung tâm để quản lý và lưu trữ dữ liệu khuôn mặt cũng như nhật ký đăng nhập (logs) từ ứng dụng. Giải pháp này không chỉ cung cấp cái nhìn trực quan, chi tiết về danh sách người dùng đã đăng ký và lịch sử truy cập, mà còn hỗ trợ tối ưu hóa quy trình quản trị thông qua cơ chế tự động xóa dữ liệu. Nhờ đó, việc quản lý tài nguyên trở nên hiệu quả hơn, loại bỏ sự phụ thuộc vào các thao tác thủ công phức tạp.



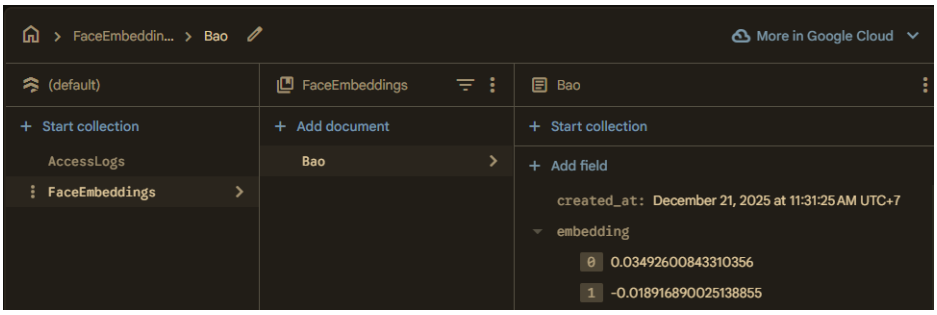
Hình 24 Trạng thái cơ sở dữ liệu trước khi có truy cập



Hình 25 Kết quả ghi log sau khi người dùng đăng nhập thành công



Hình 26 Kho lưu trữ dữ liệu khuôn mặt trước khi đăng ký



Hình 27 Dữ liệu Face Embedding được sinh ra sau khi đăng ký thành công

TÀI LIỆU THAM KHẢO

1. TOPDev. (2024). Python là gì? Tổng hợp kiến thức cho người mới bắt đầu. Truy cập ngày 4/1/2025, từ: https://topdev.vn/blog/python-la-gi/

2. fptshop. (2024). Python là gì? Tổng hợp tất tần tật kiến thức về ngôn ngữ Python có thể bạn chưa biết. Truy cập ngày 4/1/2025, từ: https://fptshop.com.vn/tin-tuc/danh-gia/python-la-gi-168825

3. VIBLO. (2019). Hướng dẫn toàn diện về Trí tuệ nhân tạo với Python, từ: https://viblo.asia/p/huong-dan-toan-dien-ve-tri-tue-nhan-tao-voi-python-dichp1-eW65G8nxKDO

4. xaydungso. (2024). AI in Python Code: Hướng Dẫn Toàn Diện Từ Cơ Bản Đến Nâng Cao, từ: https://xaydungso.vn/blog5/ai-in-python-code-vi-cb.html

5. Thai-Viet Dang, Hoai-Linh Tran. (2023). A Secured, Multilevel Face Recognition based on Head Pose Estimation, MTCNN and FaceNet. DOI: https://doi.org/10.18196/jrc.v4i4.18780

6. InternData. (2024). Facial Recognition là gì? Cách hoạt động & Ứng dụng (Có ví dụ), từ: https://interdata.vn/blog/facial-recognition-la-gi/

7. VIBLO. (2021). Nhận diện khuôn mặt với mạng MTCNN và FaceNet, từ: https://viblo.asia/p/nhan-dien-khuon-mat-voi-mang-mtcnn-va-facenet-phan-1-Qbq5QDN4lD8

8. Admin. (2024). Firebase là gì? Ưu nhược điểm & Các dịch vụ của Firebase, từ: https://fptcloud.com/firebase-la-gi/

9. TOPDev. (2024). Firebase là gì? Tìm hiểu tính năng và ưu nhược điểm của Firebase, từ: https://topdev.vn/blog/firebase-la-gi/

10. Thảo Uyên. (2024). Firebase là gì? Tổng quan và cách sử dụng Firebase hiệu quả, từ: https://cellphones.com.vn/sforum/firebase-la-gi

11. TAIPHAMEM PC. (2024). Thunkable là gì? Một số câu hỏi về về AI này bạn nên biết. từ: https://taiphanmempc.com/thunkable-la-gi-mot-so-cau-hoi-ve-ve-ai-nay-ban-nen-biet/

12. Thunkable. (2025). Thunkable Docs, từ: https://docs.thunkable.com/

13. aicandy. (2024). Tìm hiểu mô hình FaceNet cho bài toán nhận diện khuôn mặt, từ: https://aicandy.vn/tim-hieu-mo-hinh-facenet-cho-bai-toan-nhan-dien-khuon-mat/

14. VIBLO. (2021). HTTP Request là gì? Các phương thức HTTP request- https://viblo.asia/p/http-request-la-gi-cac-phuong-thuc-http-request-6J3Zgy6A5mB

15. xaydungso. (2024). HTTP Request Là Gì? Tìm Hiểu Chi Tiết Về Các Yêu Cầu HTTP Trong Lập Trình Web, từ: https://xaydungso.vn/blog/khai-niem-http-request-la-gi-va-tinh-nang-cua-no-vi-cb.html

16. LANIT. (2023). HTTP Request là gì? Cấu trúc và phương thức của HTTP Request, từ: https://lanit.com.vn/http-request-la-gi.html